ILLINOIS INSTITUTE OF TECHNOLOGY

MOVIE-LENS ANALYSIS

CSP 571 Data Preparation and Analysis

Final Report

Ranganathan Karpagam Arumuga (A20403080)
Gokul Monday Ramesh (A20401522)
Yudong Wu (A20405374)

ABSTRACT

Number of movies are released every week. There is a large amount of data related to the movies is available over the internet, because of that much data available, it is an interesting data mining topic. We have used the dataset from Movie Lens site to accomplish our objective. We train models with multiple methods such as Linear regression, logistic regression. Then we build models that can predict the movies popularity. This will give us an insight about how the people's liking for the different movie genres change over time and about the strength of association between trends in between different movie genres, insights possibly useful for the critics and other interesting insight in the history of cinematography.

1. PROBLEM DEFINITION

1.1 Introduction

MovieLens was created by the GroupLens Research, a research lab in the Department of Computer Science and Engineering at the University of Minnesota to collect largescale data for personalized recommendation research. MovieLens is a web-based recommender system that recommends movies to users. The data is generated from the reviews and ratings according to a one-to-five-star rating system submitted by users, together with the customized tags added to movies and the movie classification

A lot of research has been done on prediction of movies. Most of them include user ratings on different movies, whereas, some of them use social media (e.g. YouTube, Twitter etc.) for prediction. However, less work has done on using movies attributes such as crew, dates etc. to predict movies. The amount of data available about the movies over the internet makes its serious candidate for data mining, knowledge discovery and machine learning. Prediction of a movie is of great importance to industry; movie makers are still never sure about whether their movie will do business or not; when they should release the movie and how to advertise it.

It will be interesting to study some interesting insights in movie business such as what were the best movies of every decade, what were the best years for a genre, how many movies were produced every year or what cast is the ultimate movie cast.

1.2 Problem Statement

To identify patterns in Movielens dataset which will benefit Movie business operation

1.3 Goal

To build a logistic and multiple linear regression model over the movielens dataset.

Forecasting and plotting the obtained model for the upcoming year

2. DATA PREPATATION

The MovieLens datasets are widely used in education, research, and industry. They are downloaded hundreds of thousands of times each year, reflecting their use in popular press programming books, traditional and online courses, and software. These datasets are a product

of member activity in the MovieLens movie recommendation system, an active research platform that has hosted many experiments since its launch in 1997.The dataset is from the Movielens website hosted by Grouplens Research. (https://grouplens.org/datasets/movielens/). We have taken the dataset with 20 million records starting from the year 1840 till Sept. 2018

1) Dataset Format

movies.csv: movieId, title, genres

tags.csv: userId, movieId, tag

ratings.csv: userId, movieId, rating

Links: MovieId, Imdbid and tmdId

Variable Name	Description
MovieId	ID attached to each movie
Title	Title of the Movie
Genre	To which Genre does that particular movie belong
UserId	ID attached to each user
Rating	numeric rating is given between 0 to 5
Timestamp	Time at which the rating was recorded
Tag	A special description given to each movie by the user
ImdbID	ID attach to each movie that are present in IMDB site
tbdbID	ID of a movie from The Movie Database site

Table 1: Variable names and description

20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users. Includes tag genome data with 12 million relevance scores across 1,100 tags. It is a pity that the information involved in the dataset from MovieLens is too little. If we can get more information about the users and movies, for instance, the gender, age, education level of users, as well as the actors, directors, and length of movies, the prediction may become more overall and accurate.

3. DATA PREPROCESSING

The dataset we use is incomplete and not clean. Hence some of the techniques followed in the project is:

- Filling in missing values
- Handling date and time
- Dropping off unused levels

3.1 Filling in missing values

Variables such as Year were identified with missing values and stored in separate Data frames.

In Movie dataset, for missing values in Year we have replaced it with 0

3.2 Handling date and time

In order to achieve standard data formats and make further statistical analysis easier, we unifiedthe time format. For handling Timestamp we used mutate function using the format '%yyyy%mm%dd %H%M'

3.3 Separating Movie title and year

Among the data frames, Movie has both Title and year in a single feature. So, we separated and inserted them into separate columns and using it for further analysis.

Similarly, we split Genres and Tags

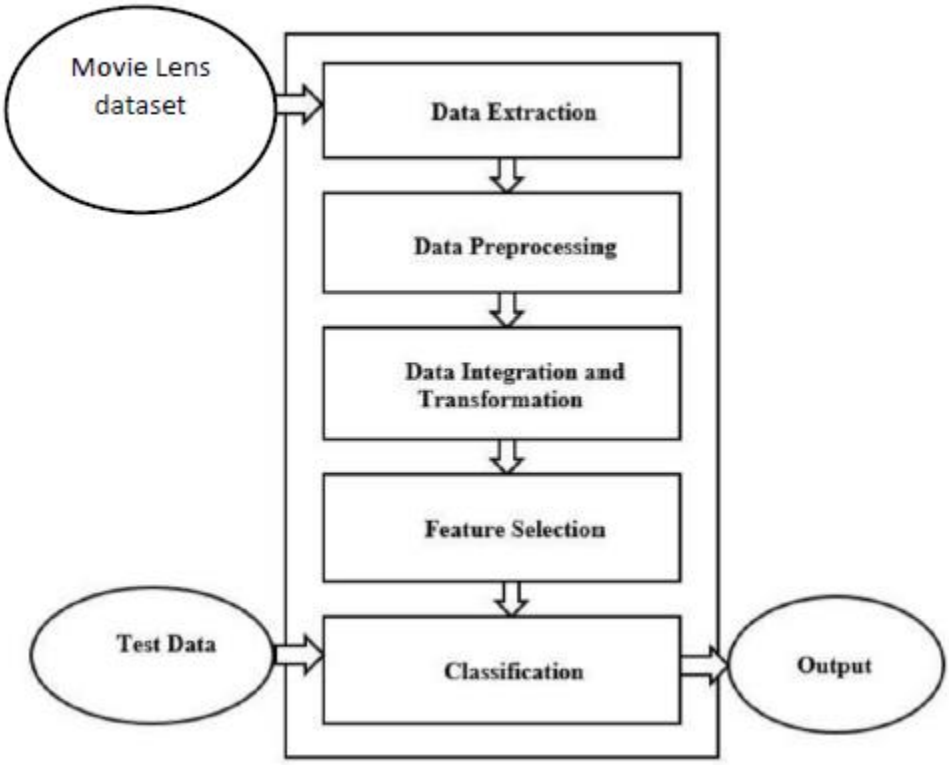


Figure 1 Steps involved in Movielens data analysis

4. EXPLANATORY DATA ANALYSIS OF DATA

Following are some of the analysis done on data before performing logistic regression:

- How many movies were produced per year?
- What were the most popular movie genres year by year?
- What tags best summarize a movie genre?
- The best movies based on users ratings?
- The best movie of a decade based on score (weighted average)
- What were the best years for a genre (based on users ratings)?

4.1 How many movies were produced per year

The following graph will show the number of movies that were produced every year starting from 1900. By studying the historic movie data, we can learn about the growth of the movie business since the start of cinematography.

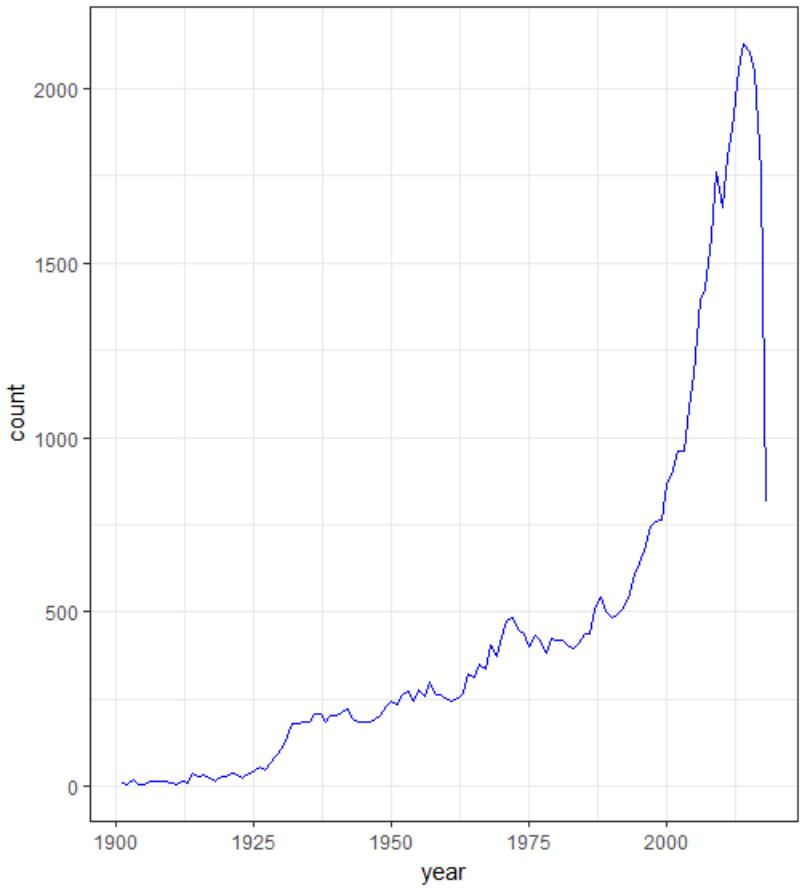


Figure 2: Total number of movies release per year

- From the graph we can infer that the movie industry flourished after 2000.
- The exponential growth in movie business is somewhat linked to the beginning of information age.
- Growing popularity of Internet must have had a positive impact on the demand for movies.

4.2 What were the popular movie genres year by year

We know how many movies were produced, but it will be interesting to see what genres were popular. We might expect that some events in history might have influenced the movie creators to produce specific genres. First, we will check what genres are the most popular in general

```
> knitr::kable(head(genres_df, 20))
```

genres	number
Drama	24144
Comedy	15956
Thriller	8216
Romance	7412
Action	7130
Horror	5555
Documentary	5118
Crime	5105
Adventure	4067
Sci-Fi	3444
Mystery	2773
Children	2749
Animation	2663
Fantasy	2637
War	1820
Western	1378
Musical	1113
Film-Noir	364
IMAX	197

```
>
```

Figure 3 Popular genres

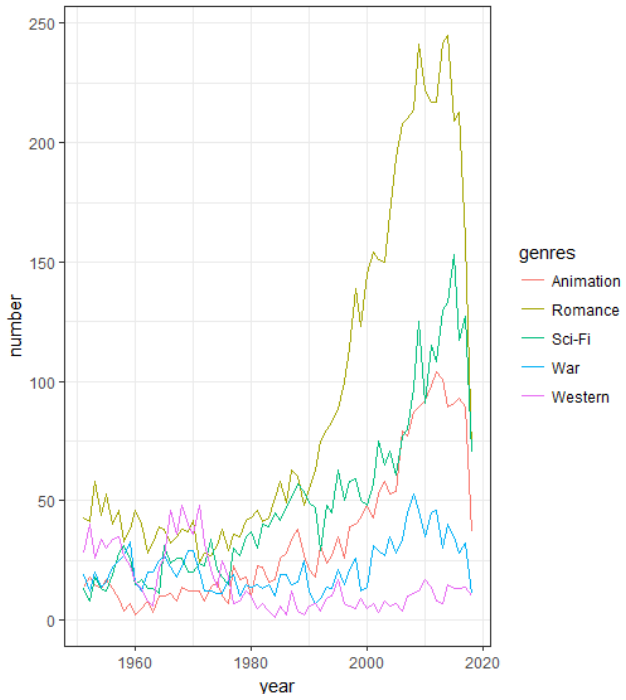


Figure 4 Comparison of Animation, Romance, Sci-Fi, War and Western genres over the years.

- I have plotted Animation, Romance, Sci-Fi, War and Western genre of movies starting from 1950.
- Romance/Drama/Comedy have always topped the chart since the start of cinematography. Significant Observations:
  - *"Western" genre between 1960 and 1980*
  - *"War" genre between 1940-1960 and 2008-2012*
  - *"Animation" between 1994-2014.*

First, we can notice a rapid growth of sci-fi movies shortly after 1969, the year of the first Moon landing. Secondly, we notice high number of westerns in 1950s and 1960s that was the time when westerns popularity was peaking. Next, we can see the rise of popularity of animated

movies, the most probable reason might be the computer animation technology advancement which made the production much easier. War movies were popular around the time when big military conflicts occurred - World War II, Vietnam War and most recently War in Afghanistan and Iraq. It's interesting to see how the world of cinematography reflected the state of the real world.

4.3 Which tags best summarize a move genre?

Looking at how each movie genre is tagged by users is a great way to see if a movie genre can be described using just a few words. Here I have created word cloud of two famous genre (i.e.) Drama and Comedy. The larger words imply that the tag was repeatedly used by many users to describe movie.

Figure 5: Word Cloud of "Drama" genre

Figure 6: Word Cloud of "Comedy" genre

As we can see some interesting keywords, but also actor and director names which indicates who is associated with the whole genre.

4.4 Finding the best movies based on user rating

We tried finding the best movies based on average rating.

movieId	title	year	count	mean	min	max
27914	Hijacking Catastrophe: 9/11, Fear & the Selling of American Empire	2004	1	5	5	5
92783	Latin Music USA	2009	1	5	5	5
93967	Keeping the Promise (Sign of the Beaver, The)	1997	1	5	5	5
94972	Best of Ernie and Bert, The	1988	1	5	5	5
95977	Junior Prom	1946	1	5	5	5
98437	Bed of Roses	1933	1	5	5	5
99450	Sun Kissed	2012	1	5	5	5
103143	Donos de Portugal	2012	1	5	5	5
103472	Alive Day Memories: Home from Iraq (Occupation Iraq)	2007	1	5	5	5
105191	Rocaterrania	2009	1	5	5	5

> |

Figure 7: Top 10 movies on all time based on average rating

This doesn't look good since our ranking has been polluted by movies with low count of reviews. To deal with this issue we are going to use the weighted average used on IMDB site for their top 250 ranking.

WR represents Weighted Average = (v/(v+m))*R + (m/(v+m))*C

- R = average for the movie (mean) = (Rating)
- v = number of votes for the movie = (votes)
- m = minimum votes required to be listed in the Top 250
- C = the mean vote across the whole report

movieId	title	year	count	mean	min	max	wr
318	Shawshank Redemption, The	1994	97999	4.424188	0.5	5	0.9949238
356	Forrest Gump	1994	97040	4.056585	0.5	5	0.9948739
296	Pulp Fiction	1994	92406	4.173971	0.5	5	0.9946182
593	Silence of the Lambs, The	1991	87899	4.151412	0.5	5	0.9943438
2571	Matrix, The	1999	84545	4.149695	0.5	5	0.9941208
260	Star Wars: Episode IV - A New Hope	1977	81815	4.120455	0.5	5	0.9939258
480	Jurassic Park	1993	76451	3.665034	0.5	5	0.9935024
527	Schindler's List	1993	71516	4.257502	0.5	5	0.9930571
110	Braveheart	1995	68803	4.008481	0.5	5	0.9927853
1	Toy Story	1995	68469	3.886649	0.5	5	0.9927504

Figure 8: Top 10 movies on all time based on weighted Average

4.5 Best movie of a decade based on Weighted average

From this graph we will be able to find the best movie of a decade since the beginning of cinematography.

decade	title	wr	mean	count
1870	Passage de Venus	0.0196078	2.550000	10
1880	Traffic Crossing Leeds Bridge	0.0253411	2.153846	13
1890	Monkeyshines, No. 1	0.0157480	2.000000	8
1900	The Kiss	0.0439771	2.934783	23
1910	Frankenstein	0.0530303	3.160714	28
1920	Cabinet of Dr. Caligari, The (Cabinet des Dr. Caligari., Das)	0.7853156	3.903499	1829
1930	All Quiet on the Western Front	0.8524203	3.936115	2888
1940	Pinocchio	0.9693871	3.446094	15833
1950	Cinderella	0.9579372	3.538948	11387
1960	Psycho	0.9797849	4.061855	24234
1970	M*A*S*H (a.k.a. MASH)	0.9658026	3.873522	14121
1980	Star Wars: Episode V - The Empire Strikes Back	0.9924610	4.133481	65822
1990	Dances with Wolves	0.9904538	3.740338	51877
2000	Gladiator	0.9898304	3.956335	48666
2010	Inception	0.9880881	4.162990	41475

Figure 9: Best movie of a decade based on weighted Average

- The above graph shows the best movie of every decade since the beginning of cinematography.
- The disadvantage of weighted ratings - low score for old movies. That's not necessarily caused by movies quality, rather small number of viewers.

4.6 Best years for a genre based on user rating:

- It seems that most of the movie genres are getting better and better.
- Also, we can see that there has been many "War" genre movies with average user ratings in the past few years
- The surge in Animation and Sci-Fi shows how well our Animation and VFX technology have been flourishing resulting in better user rating.
- Movies based on Westernization has been reduced and have got less user rating in the past decade.

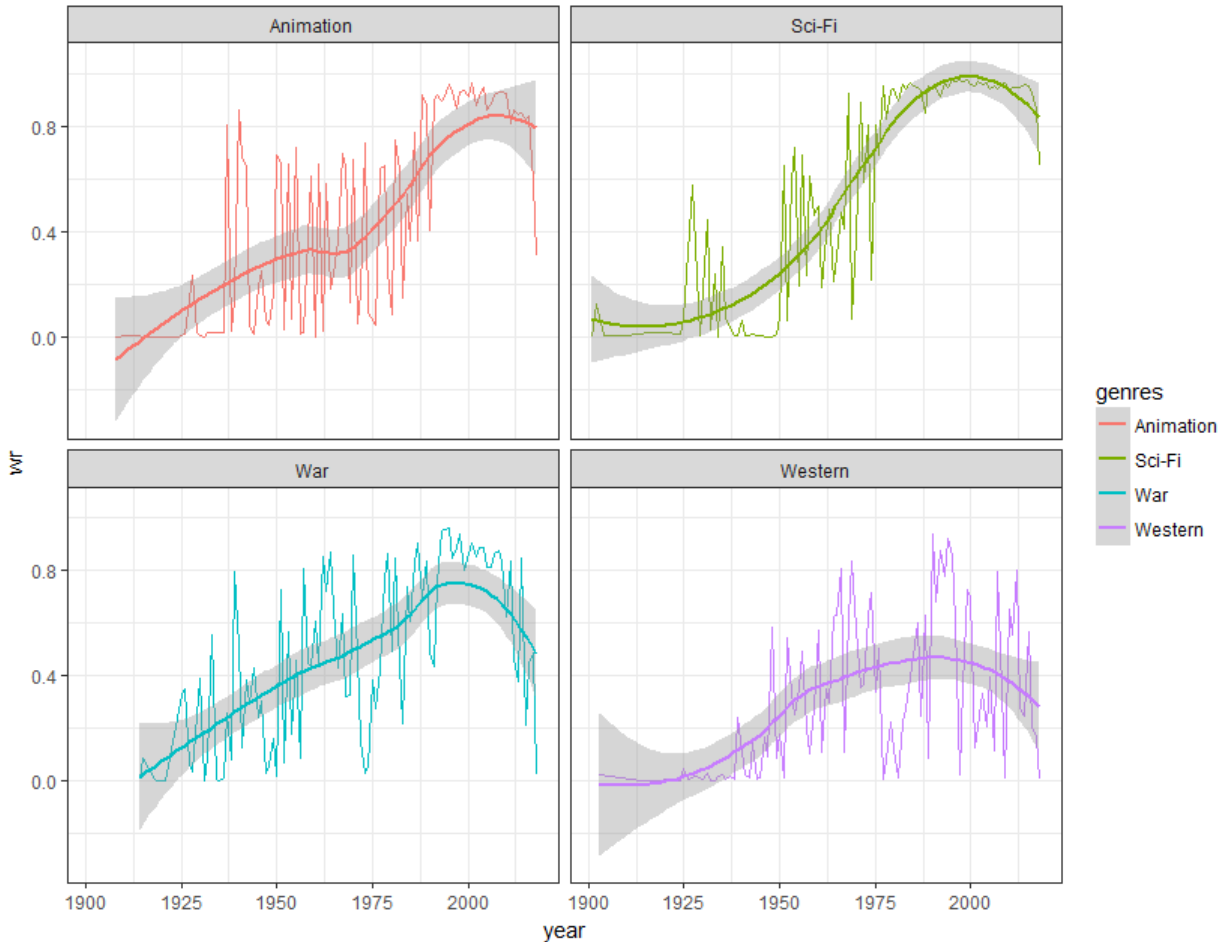


Figure 10: Best years for a genre based on users rating

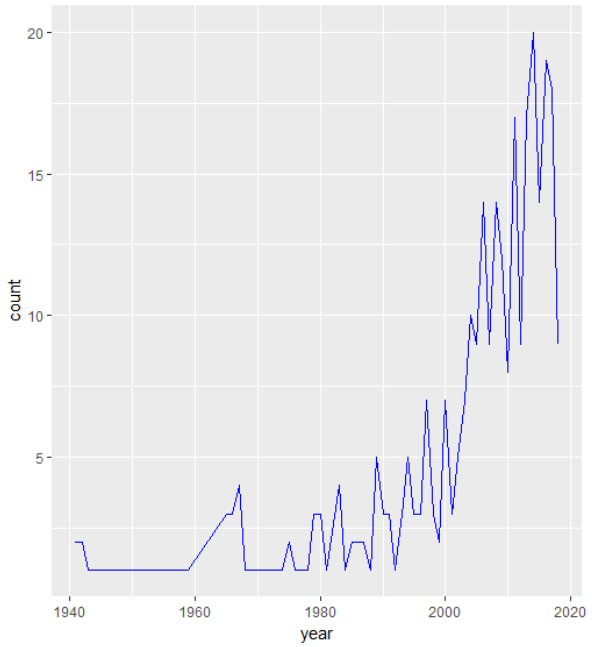
4.7 Identifying "Superhero" movie genre

After performing careful analysis on movie dataset, we can see that there isn't a specific genre for superhero movies. So, we decided to extract superhero movies based on Tags given by users. We have successfully identified "superhero" genre movies and displayed a graph below displaying the number of superhero movies released every year.

From the graph we can see that there weren't any superhero movies before 1940. And the first movie was released in 1942.

movieId	title	year	count	mean	min	max	wr
2571	Matrix, The	1999	84545	4.149695	0.5	5	0.9941208
1210	Star Wars: Episode VI - Return of the Jedi	1983	66023	3.986043	0.5	5	0.9924838
4993	Lord of the Rings: The Fellowship of the Ring, The	2001	61883	4.097943	0.5	5	0.9919850
5952	Lord of the Rings: The Two Towers, The	2002	56696	4.074705	0.5	5	0.9912581
592	Batman	1989	54448	3.386488	0.5	5	0.9909005
58559	Dark Knight, The	2008	44741	4.173756	0.5	5	0.9889481
367	Mask, The	1994	38699	3.181400	0.5	5	0.9872446
153	Batman Forever	1995	38647	2.902062	0.5	5	0.9872276
3793	X-Men	2000	36030	3.564169	0.5	5	0.9863126
5349	Spider-Man	2002	32068	3.489008	0.5	5	0.9846475

Figure 11: Top 10 superhero movies of all time based on user rating



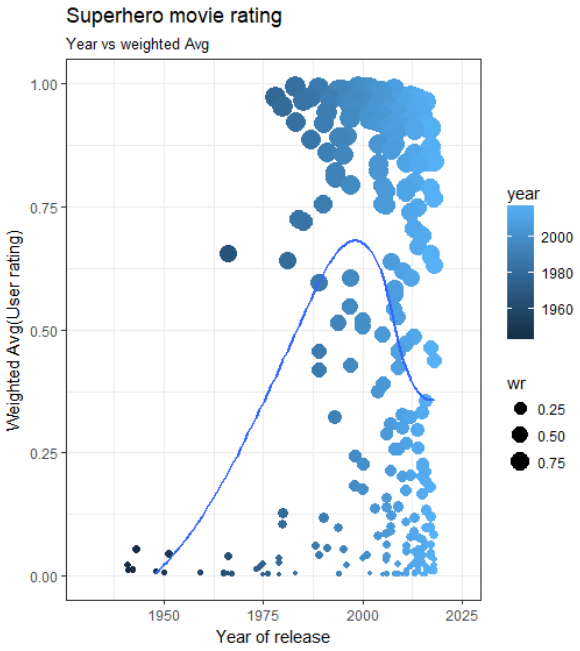


Figure 12: Number of superhero movies released per year

Figure 13: Superhero movies rating throughout the years

As we can see Both the number of movies and rating for movies have increased dramatically after 1975.

5. Linear and Logistic regression analysis

5.1 Predicting number of movies releasing next year

- To predict the number of movie that will be releasing next year we are using linear model.
- Since there are only two features (year, count) Linear model serves the best for prediction.
- There have been 816 movies till Sept. 2018 based on our Movielens dataset.
- From the linear model we can predict that the total number of movies that will be releasing in 2018 will be equal to 1150 approximately.

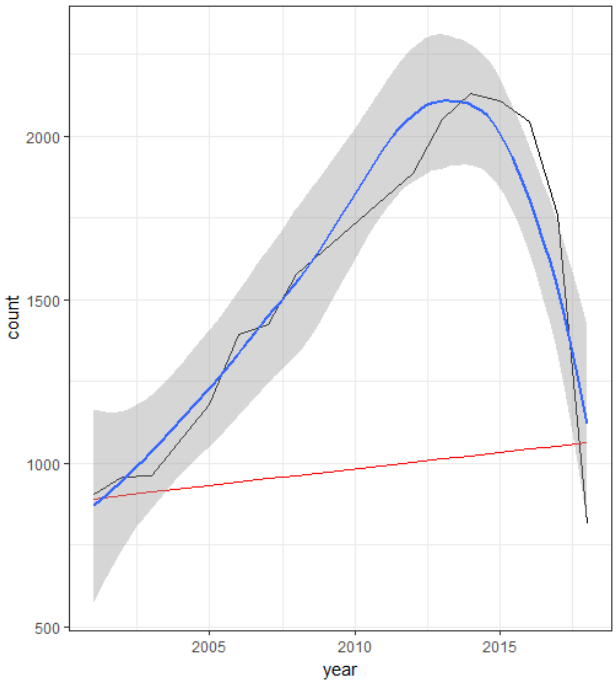


Figure 14: Movies releasing next year

5.2 Predicting if a genre is popular

- Here we have use logistic model to predict whether a genre is popular or not.
- We have set the popularity condition as Number >5000.
- The Yes, No at the top of bar implies the predicted values.

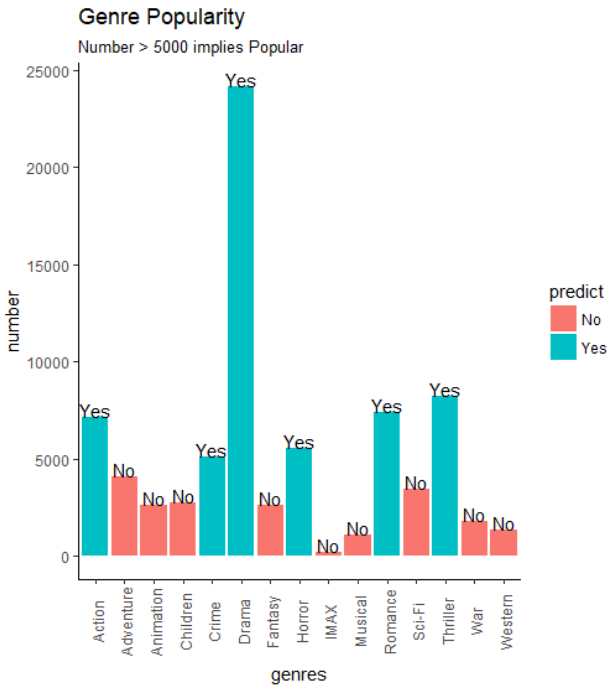


Figure 15: Bar-chart predicting if a genre is popular or not

5.3 Predicting a specific genre popularity in next year & will it be a pop(Drama) year?

- Here we have used Logistic regression model to predict whether a particular Genre will be popular in next year and will it be a “Drama” genre.
- The label “yes”, “no” at the top of bar are the predicted values.
- From this graph we can see that 2018 will be a year of “Drama” genre.

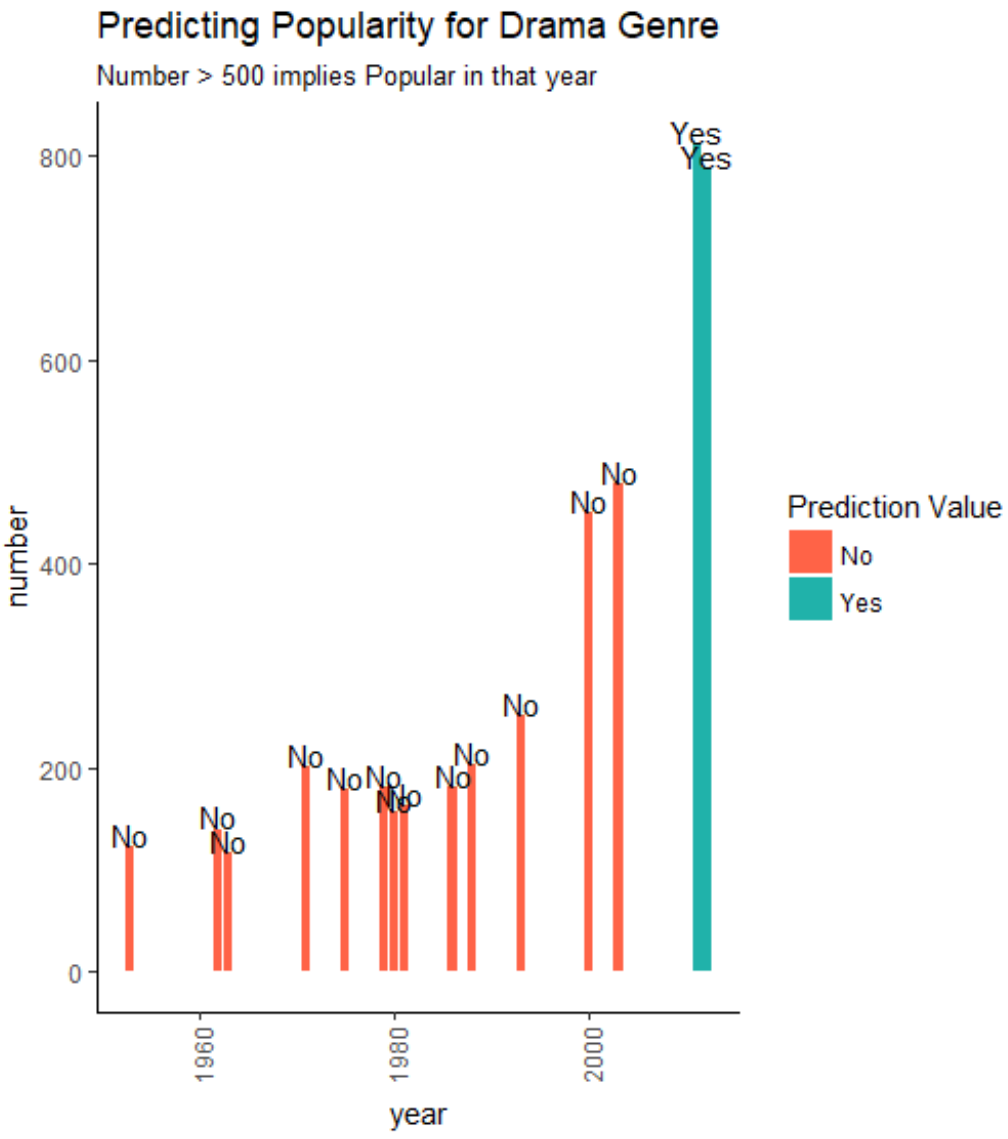


Figure 16: Bar-chart to predict if the next year will be” Drama” genre

6. CONCLUSION

It is quite interesting to see how the world of cinematography reflected the state of real world. Analyzing the MovieLens dataset gave many interesting insights into the movie business. Although it is mainly used for recommendation systems we were still able to extract some trends in the data. With web scraping methods the dataset could be easily extended to provide even more interesting observations. Overall, it was an interesting dataset to analyze that allowed using even more interesting R packages & features.

7. FUTURE WORK

We are planning to extend this project by involving movie cast through Web scraping by which we can predict some interesting scenarios such as

- *What cast is the ultimate movie cast*
- *Will the movie budget affect its score?*

8. REFERENCES

[1] Darin Im and Minh Thao Nguyen : "PREDICTING BOXOFFICE SUCCESS OF MOVIES IN THE U.S. MARKET ", CS 229, Fall

[2] 2011https://en.wikipedia.org/wiki/Film , Accessed on August 1st, 2015

[3] https://en.wikipedia.org/wiki/Internet_Movie_Database , Accessed on August 1st, 2015

[4] 'Prediction of Movies popularity Using Machine Learning Techniques', Muhammad Hassan Latif†, Hammad Afzal. National University of Sciences and technology, H-12,ISB.

9. APPENDIX

9.1 Source code

```
library(car)
library(readr)
library(caret)
library(tidyverse)
library(lubridate)
library(stringr)
library(rvest)
library(XML)
library(tidytext)
library(wordcloud)
library(data.table)
```

```
library(doParallel)
registerDoParallel()
sessionInfo()

set.seed(1234)
setwd("E:/DPA/Project/20Mil")

##################
#Loading Dataset
##################
movies <- read.csv(file="movies.csv", header=TRUE, sep=",")
ratings <- read.csv(file="ratings.csv", header=TRUE, sep=",")
links <- read.csv(file="links.csv", header=TRUE, sep=",")
tags <- read.csv(file="tags.csv", header=TRUE, sep=",")

#####################
#Formatting timestamp column in rating dataset
#####################

glimpse(ratings)
ratings_df <- ratings %>%
  mutate(timestamp = as_datetime(timestamp))

summary(ratings_df)
na_ratings_df <- ratings_df %>%
  filter(is.na(rating) | is.na(userId) | is.na(movieId) | is.na(timestamp))
summary(na_ratings_df)
###################
##Separating Year from Movie names
###################
glimpse(movies)
movies_df <- movies %>%
  # trim whitespaces
  mutate(title = str_trim(title)) %>%
  # split title to title, year
  extract(title, c("title_tmp", "year"), regex = "^(.*) \\(([0-9 \\-]*)\\)$", remove = F) %>%
  # for series take debut date
  mutate(year = if_else(str_length(year) > 4, as.integer(str_split(year, "-", simplify = T)[1]),
as.integer(year))) %>%
```

```
  # replace title NA's with original title
  mutate(title = if_else(is.na(title_tmp), title, title_tmp)) %>%
  # drop title_tmp column
  select(-title_tmp)  %>%
  # generic function to turn (no genres listed) to NA
  mutate(genres = if_else(genres == "(no genres listed)", `is.na<-`(genres), genres))

#################
##Check NA's
#################
na_movies <- movies_df %>%
  filter(is.na(title) | is.na(year))

knitr::kable(head(na_movies, 10))

na_movies[is.na(na_movies)] <- 0

#Seems that warnings appeared, because some of the movies do not have their debut year.
#We will ignore those movies in further analysis as there aren't many of them.

summary(movies_df)
##################
###Formatting timestamp column in tags dataset
##################

glimpse(tags)
tags_df <- tags %>%
  mutate(timestamp = as_datetime(timestamp))
summary(tags_df)
#################
###Links dataset doesn't have any missing values
#################
glimpse(links)
#################
#How many movies were produced per year?
# Number of movies per year/decade
movies_per_year <- movies_df %>%
  na.omit() %>% # omit missing values
  select(movieId, year) %>% # select columns we need
```

```
  group_by(year) %>% # group by year
  summarise(count = n())  %>% # count movies per year
  arrange(year)

###################
#There are some years that are missing, probably there were no movies produced in the early years.
#We can easily fix missing values using complete() function from the tidyr package.
###################
# fill missing years
movies_per_year <- movies_per_year %>%
  complete(year = full_seq(year, 1), fill = list(count = 0))
knitr::kable(head(movies_per_year, 10))

movies_per_year %>%
  filter(year > 1900) %>%
  ggplot(aes(x = year, y = count)) +
  geom_line(color="blue")

###################
#What were the most popular movie genres year by year?
###################

genres_df <- movies_df %>%
  separate_rows(genres, sep = "\\|") %>%
  group_by(genres) %>%
  summarise(number = n()) %>%
  arrange(desc(number))

knitr::kable(head(genres_df, 20))

genres_df <-genres_df %>% filter(genres != 'NA')

# Genres popularity per year
genres_popularity <- movies_df %>%
  na.omit() %>% # omit missing values
  select(movieId, year, genres) %>% # select columns we are interested in
  separate_rows(genres, sep = "\\|") %>% # separate genres into rows
  mutate(genres = as.factor(genres)) %>% # turn genres in factors
```

```
  group_by(year, genres) %>% # group data by year and genre
  summarise(number = n()) %>% # count
  complete(year = full_seq(year, 1), genres, fill = list(number = 0)) # add missing years/genres

genres_popularity %>%
  filter(year > 1950) %>%
  filter(genres %in% c("War", "Sci-Fi","Animation","Romance","Western")) %>%
  ggplot(aes(x = year, y = number)) +
  geom_line(aes(color=genres)) +
  scale_fill_brewer(palette = "Paired")

###################
####What tags best summarize a movie genre?
###################
# Tags for genres
genres_tags <- movies_df %>%
  na.omit() %>%
  select(movieId, year, genres) %>%
  separate_rows(genres, sep = "\\|") %>%
  inner_join(tags_df, by = "movieId") %>%
  select(genres, tag) %>%
  group_by(genres) %>%
  nest()

# plot wordcloud per genre
genre<-"Drama"
genre_words <- genres_tags %>%
  filter(genres == genre) %>%
  unnest() %>%
  mutate(tag = str_to_lower(tag, "en")) %>%
  anti_join(tibble(tag=c(tolower(genre)))) %>%
  count(tag)

wordcloud(genre_words$tag, genre_words$n, max.words = 50, colors=brewer.pal(10, "Dark2"))

###################
#What were the best movies of every decade (based on users' ratings)?
###################
```

```
# average rating for a movie
avg_rating <- ratings_df %>%
  inner_join(movies_df, by = "movieId") %>%
  na.omit() %>%
  select(movieId, title, rating, year) %>%
  group_by(movieId, title, year) %>%
  summarise(count = n(), mean = mean(rating), min = min(rating), max = max(rating)) %>%
  ungroup() %>%
  arrange(desc(mean))

knitr::kable(head(avg_rating, 10))

# R = average for the movie (mean) = (Rating)
# v = number of votes for the movie = (votes)
# m = minimum votes required to be listed in the Top 250
# C = the mean vote across the whole report
weighted_rating <- function(R, v, m, C) {
  return (v/(v+m))*R + (m/(v+m))*C
}

avg_rating <- avg_rating %>%
  mutate(wr = weighted_rating(mean, count, 500, mean(mean))) %>%
  arrange(desc(wr))

knitr::kable(head(avg_rating, 10))
top10_movies <-(head(avg_rating, 10))

####################################
# find best movie of a decade based on score
####################################
# heavily dependent on the number of reviews
best_per_decade <- avg_rating %>%
  mutate(decade = year  %/% 10 * 10) %>%
  arrange(year, desc(wr)) %>%
  group_by(decade) %>%
  summarise(title = first(title), wr = first(wr), mean = first(mean), count = first(count))
knitr::kable(best_per_decade)
```

```
###################################
#What were the best years for a genre (based on users' ratings)?
###################################
genres_rating <- movies_df %>%
  na.omit() %>%
  select(movieId, year, genres)

  genres_rating_new<-genres_rating %>% inner_join(ratings_df, by = "movieId") %>%
  select(-timestamp, -userId)

  genres_rating_new2<-genres_rating_new %>%
  mutate(decade = year  %/% 10 * 10)

  glimpse(genres_rating_new2)

###################################
## 75% of the sample size
smp_size <- floor(0.2 * nrow(genres_rating_new2))

## set the seed to make your partition reproducible
set.seed(123)
train_ind <- sample(seq_len(nrow(genres_rating_new2)), size = smp_size)

train <- genres_rating_new2[train_ind, ]
test <- genres_rating_new2[-train_ind, ]

genres_rating_new3<- train %>% separate_rows(genres, sep = "\\|") %>%
  group_by(year, genres) %>%
  summarise(count = n(), avg_rating = mean(rating)) %>%
  ungroup() %>%
  mutate(wr = weighted_rating(mean, count, 1000, mean(mean))) %>%
  arrange(year)

summary(genres_rating_new3)

genres_rating_new3 %>%
  filter(year >1900 ) %>%
  filter(genres %in% c("Animation", "War", "Sci-Fi", "Western")) %>%
  ggplot(aes(x = year, y = wr)) +
```

```
  geom_line(aes(group=genres, color=genres)) +
  geom_smooth(aes(group=genres, color=genres)) +
  facet_wrap(~genres)
###################################
#Superhero genre movies
###################################
superhero_ds <- tags %>% filter((tag%like%'super hero')|(tag%like%'superhero')|(tag%like%'super-hero')|(tag%like%'superheroes'))
distinct_df = superhero_ds %>% distinct(movieId)
Superhero_movies_new <-left_join(distinct_df,movies_df,by.x = "movieId", by.y = "movieId")
Superhero_movies_new %>%  filter(genres !='NA')

Superhero_movies_TotalTable <-left_join(Superhero_movies_new,ratings_df,by.x = "movieId", by.y = "movieId")

# Number of movies per year/decade
Superhero_movies_perYear <- Superhero_movies_new %>%
  na.omit() %>% # omit missing values
  select(movieId, year) %>% # select columns we need
  group_by(year) %>% # group by year
  summarise(count = n())  %>% # count movies per year
  arrange(year)
#How many Superhero movies were released per year
Superhero_movies_perYear %>%
  filter(year > 1800) %>%
  ggplot(aes(x = year, y = count)) +
  geom_line(color="blue")
#########################################
##calculating Average rating of Superhero movies
#########################################
avg_rating_superhero <-left_join(Superhero_movies_new,avg_rating,by.x = "movieId", by.y = "movieId")

#Top 10 superhero movies of all time based on weighted average
avg_rating_superhero <- avg_rating_superhero %>%
  mutate(wr = weighted_rating(mean, count, 500, mean(mean))) %>%
  arrange(desc(wr))

knitr::kable(head(avg_rating_superhero, 10))
```

```
##############################

#Scatterplot
gg <- ggplot(avg_rating_superhero, aes(x=year, y=wr)) +
  geom_point(aes(col=year, size=wr)) +
  geom_smooth(method="loess", se=F) +
  xlim(c(1930, 2018)) +
  ylim(c(0, 1)) +
  labs(subtitle="Year vs weighted Avg",
       y="Weighted Avg(User rating)",
       x="Year of release",
       title="Superhero movie rating")

plot(gg)

##############################


#Model 1: Linear Model
# How many movies will shot in next year?
# using movies_per_year subset


#movies_per_year <- movies_per_year[-c(145), ]

set.seed(100)
sample <- createDataPartition((movies_per_year$year)>1950, p = 0.7, list = F)

mTest <- movies_per_year[sample,]
mTrain <- movies_per_year[-sample,]

#mTest <-mTest %>%  filter(year > 1950)
#mTrain <-mTrain %>%  filter(year > 1950)

mLinear <- lm(count ~., data = mTrain)

mTrain$predict <- predict(mLinear, mTrain)
mTest$predict <- predict(mLinear, mTest)
```

```
#summary(movies_per_year)

mTest <-mTest %>%  filter(year > 1900)
mTrain <-mTrain %>%  filter(year > 1900)
#plot for 2 dataset
# for train set:
ggplot(mTrain, aes(x = year, y = count)) + geom_line() + geom_line(aes(y = predict), color = "blue") + stat_smooth()
# for test set:
ggplot(mTest, aes(x = year, y = count)) + geom_line() + geom_line(aes(y = predict), color = "red") + geom_smooth()

library(forecast)
library(ggfortify)

# Model 2: Logistic Model
# is this tag is popular?
# we assaume > 5000 is popular
genres_df$bool <- ifelse(genres_df$number > 5000, 0, 1)

sample <- createDataPartition(genres_df$number, p = 0.2, list = F)

g_dfTest <- genres_df[sample,]
g_dfTrain <- genres_df[-sample,]

g_dfTrain <- g_dfTrain %>%  filter(genres !='NA')
g_dfTest <- g_dfTest %>%  filter(genres !='NA')

g_dfGlm <- glm(bool ~ number, data = g_dfTrain, family = binomial())

g_dfTrain$predict <- ifelse(predict(g_dfGlm, g_dfTrain, type = "response") > 0.5, "No", "Yes")
g_dfTest$predict <- ifelse(predict(g_dfGlm, g_dfTest, type = "response") > 0.5, "No", "Yes")

#View(g_dfTrain)
#View(g_dfTest)

ggplot(data = g_dfTrain, aes(genres, number, group = predict)) +
  geom_col(aes(fill = predict), position = "dodge") +
```

```
  geom_text(
    aes(label = predict, y = number + 0.05),
    position = position_dodge(0.9),
    vjust = 0
  ) +theme(axis.text.x = element_text(angle=90, vjust=0.6)) +
  labs(title="Genre Popularity",
       subtitle="Number > 5000 implies Popular")

ggplot(data = g_dfTest, aes(genres, number, group = predict)) +
  geom_col(aes(fill = predict), position = "dodge") +
  geom_text(
    aes(label = predict, y = number + 0.05),
    position = position_dodge(0.9),
    vjust = 0
  ) +theme(axis.text.x = element_text(angle=90, vjust=0.6)) +
  labs(title="Genre Popularity",
       subtitle="Number > 5000 implies Popular")



# Model 3: Logistic Model
# predict the specific genres popilarity in next year & will it be a pop year?
# use geres_popularity subset
drmaPop <- subset(genres_popularity, genres_popularity$genres == "Drama")
#View(drmaPop)
# set pop > 500 to pop

drmaPop$pop <- ifelse(drmaPop$number > 500, 1, 0)
sample <- createDataPartition(drmaPop$year, p = 0.2, list = F)

drampop_test <- drmaPop[sample,]
drampop_train <- drmaPop[-sample,]

#drampop_linear <- lm(number ~ year, data = drampop_train)
drmapop_glm <- glm(pop ~ year + number, data = drampop_train, family = binomial())
#pTrain <- predict(drampop_linear, drampop_train)
#pTest <- predict(drampop_linear, drampop_test)
drampop_train$pop_predict <-ifelse(predict(drmapop_glm, drampop_train) > 0.5, 1, 0)
drampop_test$pop_predict <- ifelse(predict(drmapop_glm, drampop_test) > 0.5, 1, 0)
```

```
drampop_train <- drampop_train %>%  filter(year >1950)
drampop_test <- drampop_test %>%  filter(year >1950)

# blue is predict
# for train set:

drampop_test$pop_predict <- ifelse(drampop_test$pop_predict=="0", "No", "Yes")
drampop_train$pop_predict <- ifelse(drampop_train$pop_predict=="0", "No", "Yes")

#View(drampop_test)
ggplot(data = drampop_test, aes(year, number, group = pop_predict)) +
  geom_col(aes(fill = pop_predict), position = "dodge") +
  geom_text(  aes(label = pop_predict, y = number + 0.05),position = position_dodge(0.9), vjust
= 0 ) +
  theme(axis.text.x = element_text(angle=90, vjust=0.6)) +
  labs(title="Predicting Popularity for Drama Genre", subtitle="Number > 500 implies Popular in
that year") +
  scale_fill_manual("Prediction Value", values = c("No" = "Tomato", "Yes" = "LightSeaGreen"))


ggplot(data = drampop_train, aes(year, number, group = pop_predict)) +
  geom_col(aes(fill = pop_predict), position = "dodge") +
  geom_text(  aes(label = pop_predict, y = number + 0.05),position = position_dodge(0.9), vjust
= 0 ) +
  theme(axis.text.x = element_text(angle=90, vjust=0.6)) +
  labs(title="Predicting Popularity for Drama Genre", subtitle="Number > 500 implies Popular in
that year")
```