MULTIMEDIA UNIVERSITY®

Bridging Boundaries, Connecting Minds

Faculty of Computing and Informatics (FCI)

TDS3301 DATA MINING

Assignment 1

Prepared by:

Pay Eong Ian	1122703150
Tan Suk Mei	1132700522
Chew Yoke Teng	1132701833
Loh Yun Kuen	1131122813

PART 1: EXPLORATORY DATA ANALYSIS

A. Dataset Description

For this assignment, we have chosen the "IMDB Movies Dataset" which obtained from Kaggle website. The author has scraped 14761 movies from IMDB website, spanning across 28 genres within 129 years, from year 1888 to 2017. This dataset contains 15190 rows representing observations and 44 columns representing variables. Variables such as the movie title, imdb rating, number of reviews, duration, movie genre and so on were included to allow us to easily tell the greatness of movie in order to provide movie recommendation. As this dataset contains huge amount of data, it may take a very long time to make analysis. Moreover, some of the data seems to be inconsistent and consist of missing values. Hence, we have decided to perform data reduction on the dataset to reduce the volume of the dataset and data cleaning to remove data with issues in order to produce a clean dataset.

B. Dataset Insights

The chosen dataset IMDB Movie is mainly to get insights for the **movie recommendation** to movie watchers. Based on the variables 'year' of the chosen dataset, one of the possible insights can be obtained is in which year, the **most number of movies are produced**. Next, we can get the **top rating movies** according to the variable 'imdbRating' from the chosen dataset. Besides, we can also get an observation on the number of nominations of the movies. This gives an insight that the movie with **higher number of nominations** is most welcomed by movie watchers. An insight can be obtained from the number of user reviews is **the higher the number of user reviews**, the more controversial the movie is. As a conclusion, movie recommendation can help movie watchers make decision easily based on year, movie genre, rating, number of nominations and the number of user reviews.

C. Data Mining Technique (Classification)

Classification would be relevant for this dataset. A classification model is based on a given input and predict a certain outcome. For this dataset, we will classify the imdbRating attribute in order to easily provide movie recommendation. As example, we can group the imdbRating attribute into a recommendation class such as: 8.0 - 10 (excellent), 7.0 - 7.9 (good), 5.0 - 6.9 (average), 3.0 - 4.9 (fair), 0 - 2.9 (poor). With the class in the new attribute, we can easily identify the top rated movies and provide people who love to watch movies with recommendation on films that might be worth watching.

```
#classify imdbRating into new column Recommendation
movie$Recommendation[movie$imdbRating>=0 &
movie$imdbRating<3.0]<-"Poor"
movie$Recommendation[movie$imdbRating>=3.0 &
movie$imdbRating<5.0]<-"Fair"
movie$Recommendation[movie$imdbRating>=5.0 &
movie$imdbRating<7.0]<-"Average"
movie$Recommendation[movie$imdbRating>=7.0 &
movie$imdbRating<8.0]<-"Good"
movie$Recommendation[movie$imdbRating>=8.0 &
movie$imdbRating<=10]<-"Excellent"
```

D. Data Quality Issues

There are 15190 observations and 44 variables in this dataset. The data quality issues that have been found are false values, missing values, and noisy data. Foremost, we removed the data before year 2000. We found that there are 3 unnecessary variables in the dataset which are the **fn**, **tid**, and **wordsInTitle**, due to fn and tid are actually representing the id of the movie, and "wordsInTitle" is just showing the alphabets that were extracted from the variable "title".

9 variables, which are **url**, **imdbRating**, **ratingCount**, **duration, year**, **type**, **nrOfWins**, **nrOfNominations**, and **nrOfPhotos**, were observed that containing massive of false values. For example, in variable "url", there are a lot of false values instead of links such as, a movie name in the variable "url". Likewise, there is a false value 8.1 falls under the variable "year", which is actually the imdbRating, links found under variable "ratingCount", value of year found under variable "type", value of type found under variable "nrOfWins", value of year and type found under variable "nrOfNominations", and value of type under variable "nrOfPhotos". An important part, we found that there are some different type values under the variable "type". For example, instead of *video.movie,* there are some kind of other types such as *games*, *video.tv*, and *video.episode*. Therefore, we removed observations which are not equal to movie type as we are doing the dataset about movies.

Besides, there are missing values found in 3 variables which are **imdbRating**, **ratingCount**, and **duration**. Most of them were found in the variable "imdbRating" and "ratingCount", followed by the others. Function complete.case() is used to remove NA, the missing value. The following issue is about the noisy data in column **title**. After going through the value in that particular variable, we discovered that a lot of ambiguous values were appeared in the title such as Ã, ¶, ¤, §, and others and these lead to the inconsistent of data. Thus, we removed the noisy data singly using R.

E. Pre-Processing Tasks

Refer to asg1.R for all of the data pre-processing steps.

Data Reduction

```
#remove rows with values 0 and 1
movie<-movie[movie$fn != 0,]
movie<-movie[movie$fn != 1,]

#remove rows with null values
movie<-movie[complete.cases(movie),]

#remove tv series and only select movies
movie<-movie[movie$type == "video.movie",]

#remove rows before 2000
movie<-movie[movie$year > 2000,]
```

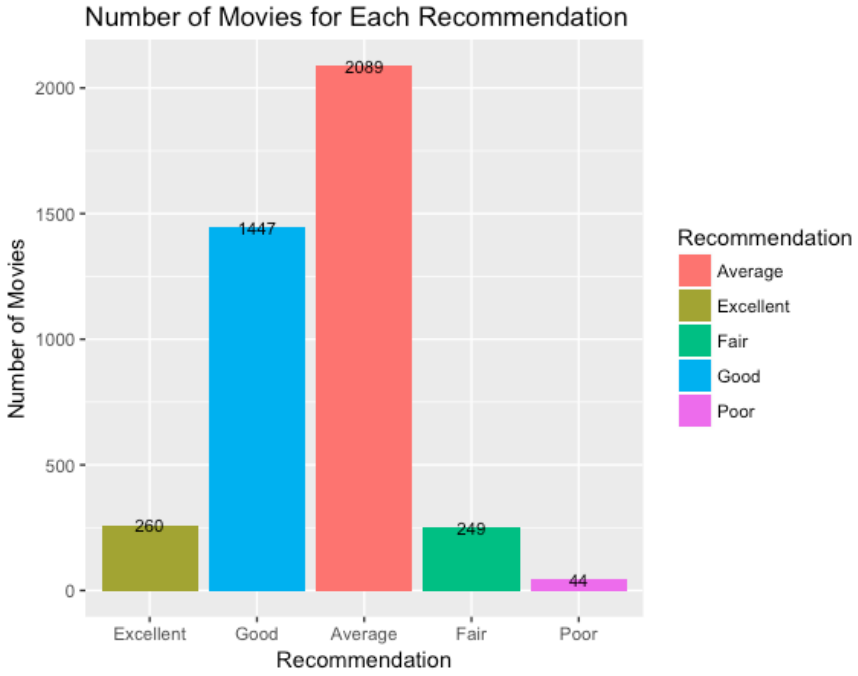
Data Cleaning

```
#remove year and strings in the brackets from title
movie$title<-gsub("\\s\\(.....*","",movie$title)

#remove noisy data in the title
movie$title<-gsub("\\Ã"," ",movie$title)
movie$title<-gsub("\\¼"," ",movie$title)
movie$title<-gsub("\\¤"," ",movie$title)
```

IDENTIFY NUMBER OF TOP RATED MOVIES TO RECOMMEND

```
recommendation = count(movie, 'Recommendation')
recommendation <- recommendation[with(recommendation,
order(-freq)),]
recommendation <- head(recommendation,5)
recommendation.count <- table(movie$imdbRating)

p<-ggplot(data=recommendation, aes(x=Recommendation,
y=freq, fill=Recommendation)) +
 geom_bar(stat="identity") + labs(title="Number of Movies
for Each Recommendation", x="Recommendation", y = "Number
of Movies") +
 geom_text(aes(label=freq), color="black", hjust=0.5,
size=3)
p + scale_x_discrete(limits=c("Excellent", "Good",
"Average", "Fair", "Poor"))
```

The graph we obtained from this code is the total count of recommendation **(frequency)** according to the imdb ratings (0-10) which have been classified into 5 categories **(Excellent, Good, Average, Fair, Poor)**.



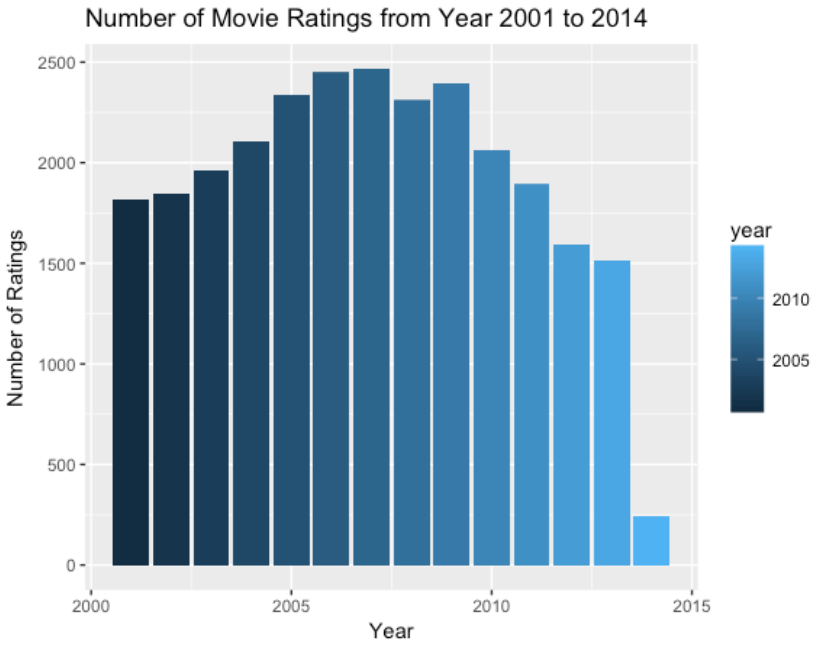
There is a total count of 2089 at **Average** of recommendation which is the highest among the five recommendations. This shows that there are 2089 numbers of movies being rated with value greater than 5 but less than 7. The recommendation with lesser total count is **Poor** with a total of 44, and the rating values of the movies fall in poor recommendations are from 0 to 2.9 only.

The second higher total recommendations count are 1447 according to the total number of movies with **Good** recommendations and rating value from 7 to 7.9. The others two recommendation, **Excellent** and **Fair** has a total count of 260 and 249 respectively. Movies which fall in these two recommendations have rating from 8 to 10 and 3 to 4.9. From the graph above, we know that there is an option of 260 top rated movies we can select from to provide recommendation for movie lovers.

IDENTIFY NUMBER OF MOVIE RATINGS FROM YEAR 2001-2014

```
p<-ggplot(movie, aes(x=year, y=imdbRating, fill=year)) +
geom_bar(stat="identity") +
 labs(title="Number of Movie Ratings from Year 2001 to
2014", x="Year", y="Number of Ratings")
```

From the code above, we obtained the graph that shows the total **Number of Ratings** for movies produced between year 2001 to 2014.

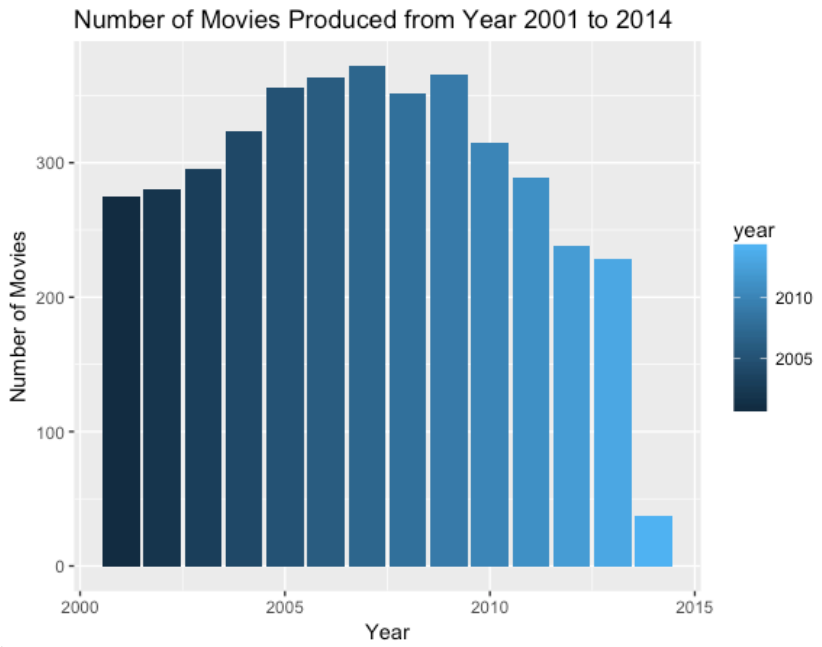


Through this graph, we can know the total number of ratings on movies in year **2007** is the highest among the years. Unfortunately, the lowest total number of movie ratings falls in year **2014**. This is due to the less production of movie in year 2014.

IDENTIFY NUMBER OF MOVIES PRODUCED FROM YEAR 2001-2014

```
year = count(movie, 'year')
year <- year[with(year, order(-freq)),]
year <- head(year,14)
year.count <- table(movie$title)

p<-ggplot(data=year, aes(x=year, y=freq, fill=year)) +
  geom_bar(stat="identity") + labs(title="Number of
Movies Produced from Year 2001 to 2014", x="Year", y =
"Number of Movies")
```

The graph we obtained from the code above shown the **Number of Movies Produced** from year 2001 to year 2014.



From the graph, we can observe that in year **2007** there is a total of 372 movies are produced. In year **2014**, there are only 47 movies is produced.