---

title: "Movies Dataset"

author: "Mehmet Kadri Gofralılar"

date: "10/15/2021"

output: html\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

library(tidyverse)

```

## Explaining the Dataset

This data was downloaded by me from <https://www.kaggle.com/ruchi798/movies-on-netflix-prime-video-hulu-and-disney>.

This dataset is an amalgamation of:

-data that was scraped, which comprised a comprehensive list of \*\*movies\*\* available on various \*\*streaming platforms\*\*

-\*\*IMDb\*\* dataset

At first, we need to read the data.

```{r load\_data}

movies <- read.csv("data/MoviesOnStreamingPlatforms\_updated.csv")

head(movies)

```

## Cleaning the Data

Now, we have to tidy the data a little bit.

```{r data\_preparation}

movies$Platform="NotDetermined"

rowCount <- nrow(movies)

for (platformColumn in 8:11) {

for (i in 1:rowCount) {

if (movies[i,platformColumn]==1) {

movies[i,18]=colnames(movies[platformColumn])

}

}

}

movies <- subset( movies, select = -Netflix )

movies <- subset( movies, select = -Hulu )

movies <- subset( movies, select = -Prime\_Video )

movies <- subset( movies, select = -Disney )

movies <- subset( movies, select = -Index )

movies <- subset( movies, select = -Type )

movie\_df <-as\_tibble(select(movies, Title, Year, Age, IMDb, Rotten.Tomatoes, Platform, Genres, Country, Language, Runtime))

movie\_df <- filter(movie\_df, (IMDb!=''))

movie\_df <- movie\_df %>%

separate(IMDb, into = c("IMDb", "imdb\_will\_be\_deleted"), sep = "/" , convert = TRUE)

movie\_df <- filter(movie\_df, (Rotten.Tomatoes!=''))

movie\_df <- movie\_df %>%

separate(Rotten.Tomatoes, into = c("Rotten.Tomatoes", "Rotten.Tomatoes\_will\_be\_deleted") , sep = "/" , convert = TRUE)

movie\_df <- subset( movie\_df, select = - imdb\_will\_be\_deleted )

movie\_df <- subset( movie\_df, select = - Rotten.Tomatoes\_will\_be\_deleted )

head(movies)

```

We've converted our dataframe to tibble, because tibbles have a refined print method that shows only the first 10 rows, and all the columns that fit on screen.

Now, lets see age restriction of movies:

```{r plot1}

ggplot(filter(movie\_df, Age!="")) +

geom\_bar(mapping=aes(x = Platform, fill= Age),position = "dodge")+

labs(

x = "Movie Platforms",

y = "Movie Counts",

title = "Age Restrictions Of Movies Made by Different Platforms")

```

By looking at this graph, we can say that

```{r plot2}

ggplot(movie\_df) +

geom\_point(mapping=aes(x = IMDb, y = Rotten.Tomatoes, color= Platform))+

labs(

x = "IMDb Ratings",

y = "Rotten Tomatoes Ratings",

title = "Ratings of Movies on Different Sites Made by Different Platforms")

```

### As we could have predicted, there is a positive correlation between attack and total points, but there are some pokemon that have really low attack points, but decent total points.

```{r plot3}

ggplot(filter(movie\_df, Year==2021)) +

geom\_point(mapping=aes(x = IMDb, y = Rotten.Tomatoes, color= Platform)) +

labs(

x = "IMDb Ratings",

y = "Rotten Tomatoes Ratings",

title = "Ratings of Movies on Different Sites Made by Different Platforms in 2021")

```

### By giving a color to each pokemon based on their hp (health points), we can see that the exceptions are pokemons with high hp.