Final project week 10

2023-08-09

Movie Recommendation System

Introduction

In the vast realm of movies, the quest for a personalized cinematic experience has led us to the challenge of crafting a robust movie recommendation system. This journey is driven by the confluence of R's analytical prowess and a rich dataset comprising variables like genre, rating, timestamp, title, userid, and movie id. Our story revolves around unraveling the enigma of individual preferences to create a cinematic symphony tailored to each viewer's unique tastes.

Problem Statement

The main statistical question was to understand the factors influencing movie ratings and uncover any underlying patterns in user preferences. And to build a comprehensive movie recommendation system capable of offering personalized movie suggestions to users. I also aimed to investigate whether certain genres or movie release years had a significant impact on movie ratings.

```r
# Remove the last 6 characters from the 'column_name' column and create a new column
movies$New_Title <- substr(movies$title, 1, nchar(movies$title)-6)
```

```r
movies$Year <- substr(movies$title, nchar(movies$title)-5, nchar(movies$title))
movies$Year <- gsub("[()]", "", movies$Year)
movies$Year <- as.numeric(movies$Year, errors = "coerce")
```

```
## Warning: NAs introduced by coercion
```

```r
na.omit(movies)
```

```
## # A tibble: 9,733 x 5
##    movieId title                              genres             New_Title  Year
##      <dbl> <chr>                              <chr>              <chr>     <dbl>
##  1       1 Toy Story (1995)                   Adventure|Animati~ "Toy Sto~  1995
##  2       2 Jumanji (1995)                     Adventure|Childre~ "Jumanji~  1995
##  3       3 Grumpier Old Men (1995)            Comedy|Romance     "Grumpie~  1995
##  4       4 Waiting to Exhale (1995)           Comedy|Drama|Roma~ "Waiting~  1995
##  5       5 Father of the Bride Part II (1995) Comedy             "Father ~  1995
##  6       6 Heat (1995)                        Action|Crime|Thri~ "Heat "    1995
##  7       7 Sabrina (1995)                     Comedy|Romance     "Sabrina~  1995
##  8       8 Tom and Huck (1995)                Adventure|Children "Tom and~  1995
##  9       9 Sudden Death (1995)                Action             "Sudden ~  1995
## 10      10 GoldenEye (1995)                   Action|Adventure|~ "GoldenE~  1995
## # i 9,723 more rows
```

```r
merged_df <- merge(movies, ratings, by = "movieId") #%>% select(-title)
#na.omit(merged_df)
head(merged_df, 5)
```

```
##   movieId            title                                      genres
## 1       1 Toy Story (1995) Adventure|Animation|Children|Comedy|Fantasy
## 2       1 Toy Story (1995) Adventure|Animation|Children|Comedy|Fantasy
## 3       1 Toy Story (1995) Adventure|Animation|Children|Comedy|Fantasy
## 4       1 Toy Story (1995) Adventure|Animation|Children|Comedy|Fantasy
## 5       1 Toy Story (1995) Adventure|Animation|Children|Comedy|Fantasy
##    New_Title Year userId rating  timestamp
## 1 Toy Story  1995    590    4.0 1258420408
## 2 Toy Story  1995    328    5.0 1494210665
## 3 Toy Story  1995    232    3.5 1076955621
## 4 Toy Story  1995    608    2.5 1117408267
## 5 Toy Story  1995    448    5.0 1019126661
```

```r
summary(ratings)
```

```
##      userId         movieId           rating        timestamp
##  Min.   :  1.0   Min.   :     1   Min.   :0.500   Min.   :8.281e+08
##  1st Qu.:177.0   1st Qu.:  1199   1st Qu.:3.000   1st Qu.:1.019e+09
##  Median :325.0   Median :  2991   Median :3.500   Median :1.186e+09
##  Mean   :326.1   Mean   : 19435   Mean   :3.502   Mean   :1.206e+09
##  3rd Qu.:477.0   3rd Qu.:  8122   3rd Qu.:4.000   3rd Qu.:1.436e+09
##  Max.   :610.0   Max.   :193609   Max.   :5.000   Max.   :1.538e+09
```

```r
summary(movies)
```

```
##     movieId          title              genres           New_Title
##  Min.   :     1   Length:9737        Length:9737        Length:9737
##  1st Qu.:  3247   Class :character   Class :character   Class :character
##  Median :  7294   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 42159
##  3rd Qu.: 76143
##  Max.   :193609
##
##       Year
##  Min.   :1902
##  1st Qu.:1988
##  Median :1999
##  Mean   :1995
##  3rd Qu.:2008
##  Max.   :2018
##  NA's   :4
```

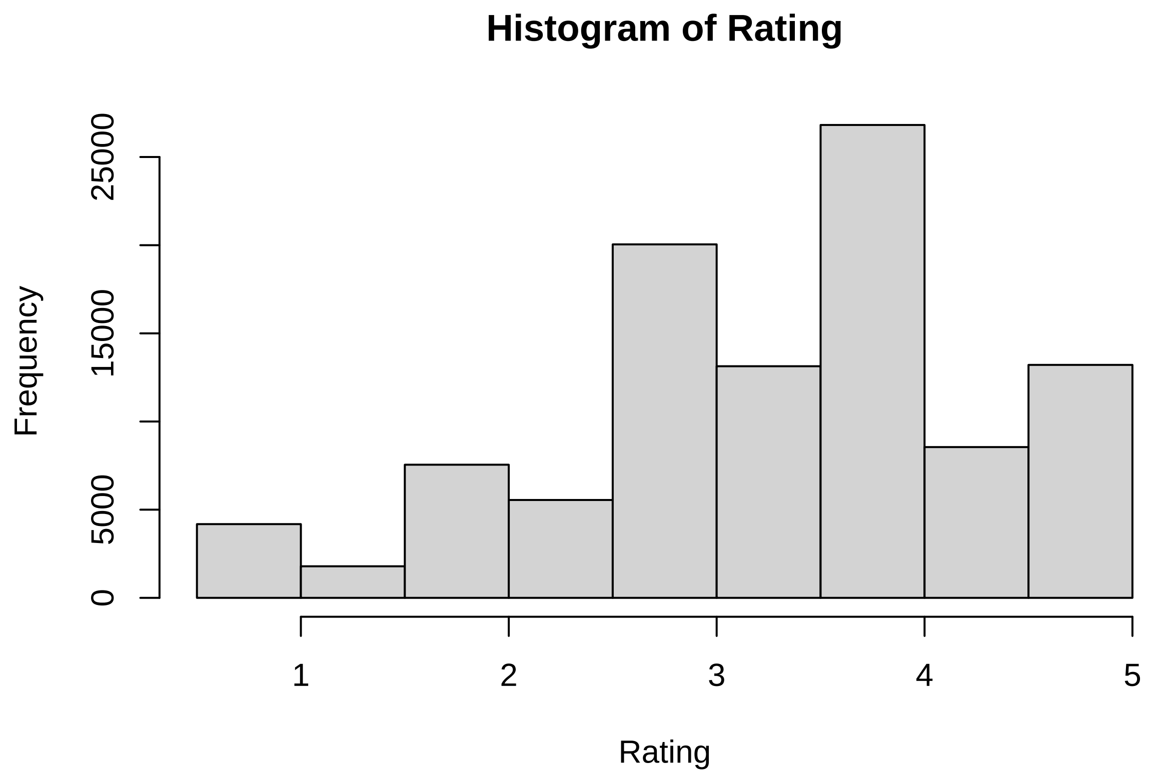
Approach

To tackle the problem statement, two crucial datasets were curated: ‘movies’ containing movie attributes and ‘ratings’ with user ratings and reviews. Armed with the ‘tidyverse’ and ‘tidytext’ packages, I embarked on data wrangling, visualization and correlation, and regression models to gain an in-depth understanding of the data. The reccommenderlab library was also used to obtain the top 10 movie recommendations for user1.

```r
library(ggplot2)
library(scales)
```

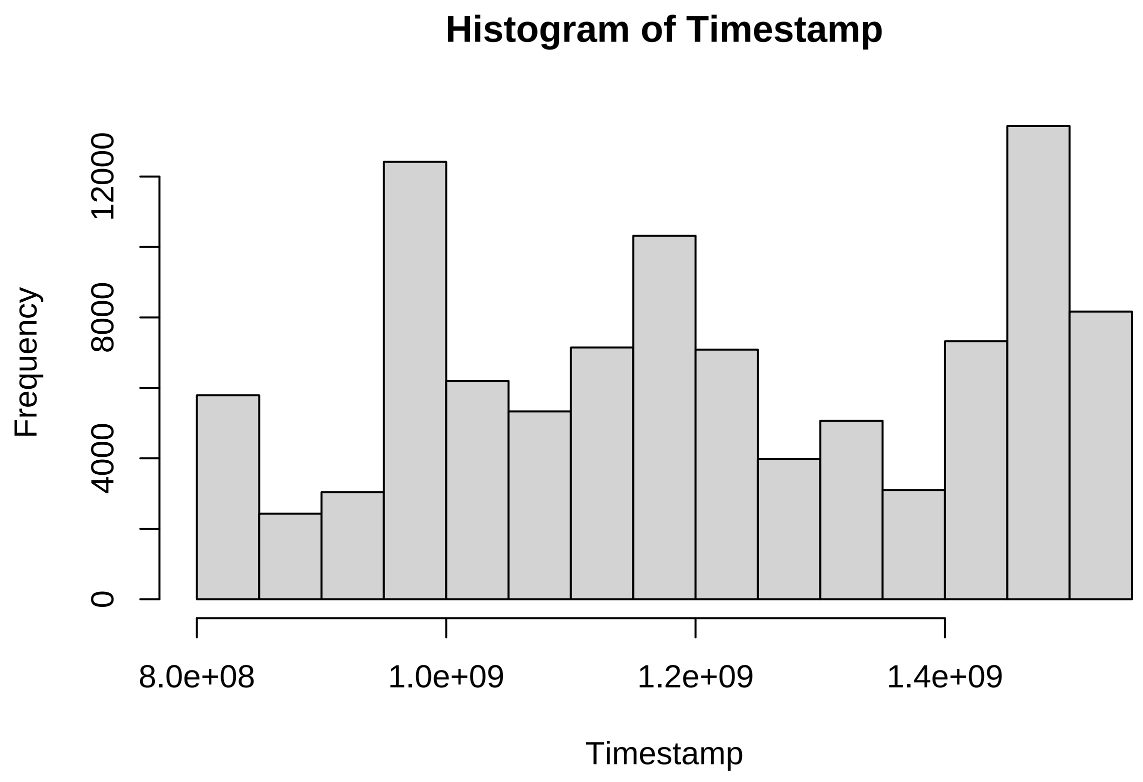
```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
# Plot the histogram
hist(ratings$rating, breaks = 10, border = "black", xlab="Rating",
ylab="Frequency",main="Histogram of Rating")
```

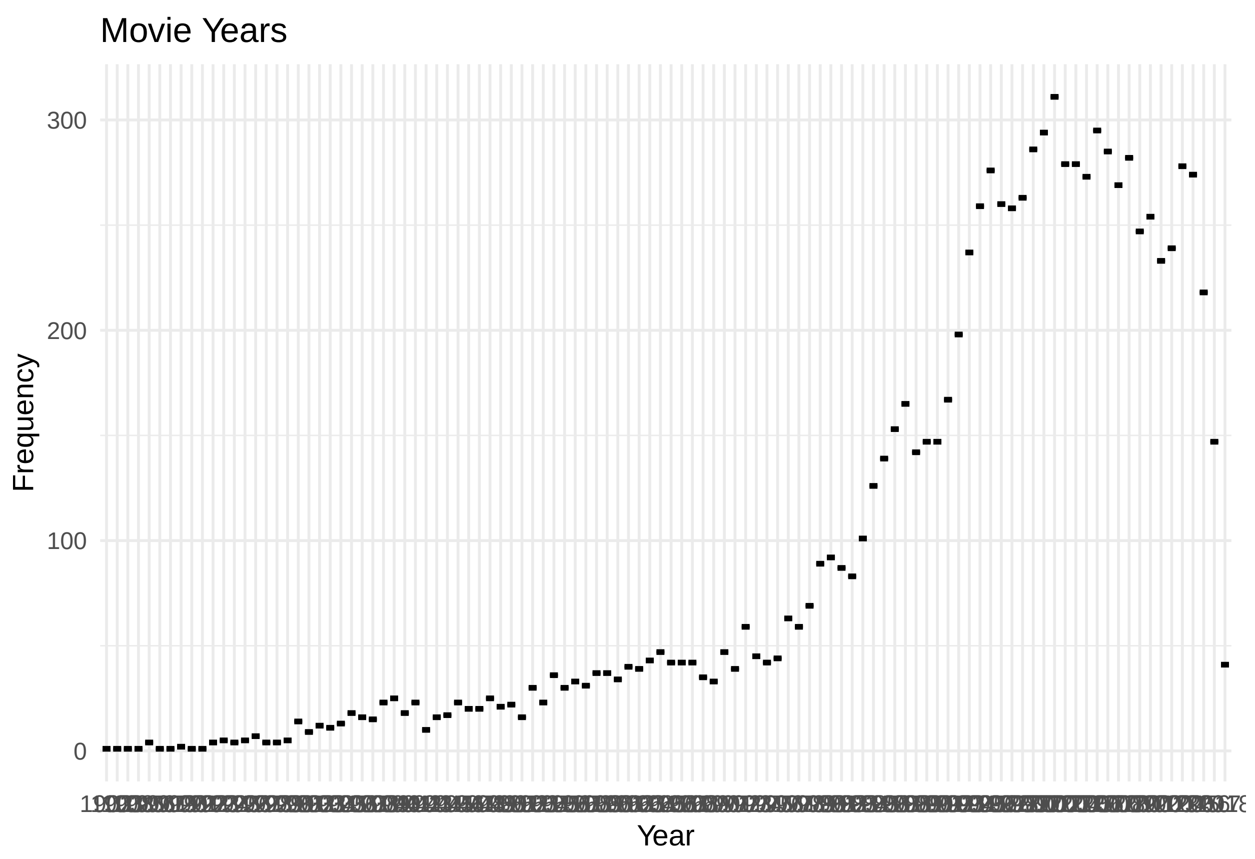


```r
hist(ratings$timestamp, breaks = 20, border = "black", xlab="Timestamp",
ylab="Frequency",main="Histogram of Timestamp")
```

```
## Warning in breaks[-1L] + breaks[-nB]: NAs produced by integer overflow
```



```r
# Calculate the frequency of each movie year
year_count <- table(movies$Year)

# Convert the result to a data frame for  plotting
year_count_df <- data.frame(Year = as.numeric(names(year_count)), Frequency = as.numeric(year_count))


boxplot <- ggplot(year_count_df, aes(x = factor(Year), y = Frequency)) +
  geom_boxplot(fill = "skyblue", color = "black") +
  labs(x = "Year", y = "Frequency", title = "Movie Years") +
  theme_minimal()

# Display the boxplot
print(boxplot)
```

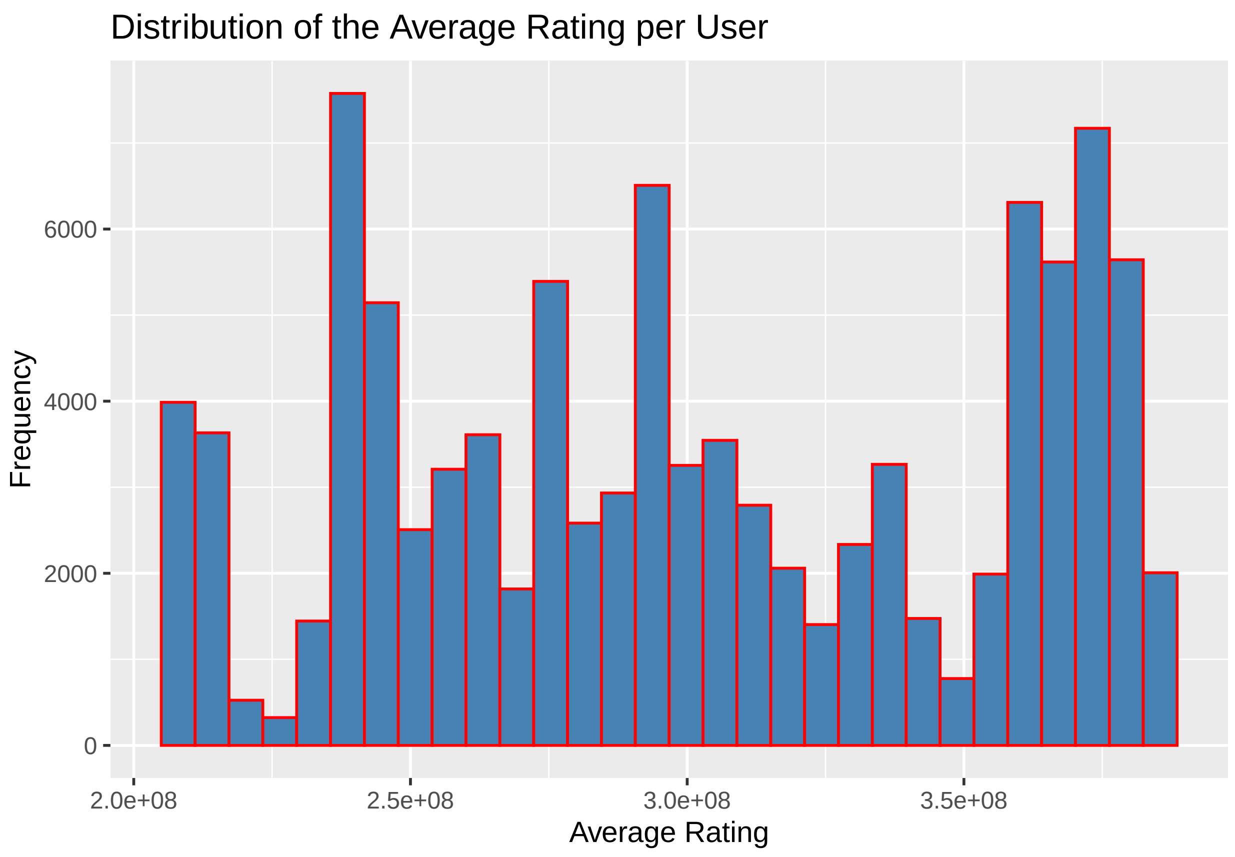


```
# # Calculate the frequency of each genre
# genres_count <- table(movies$New_Title)
#
# scatter_plot <- ggplot(data = as.data.frame(genres_count), aes(x = as.factor(Var1), y = Freq)) +
#   geom_point(size = 3, color = "blue") +
#   labs(x = "Genres", y = "Frequency", title = "Scatter Plot of Genres") +
#   theme_minimal()
#
# print(scatter_plot)
```

```
library(dplyr)
#calculate average ratings by movie id
avg_ratings <- ratings %>%
  group_by(movieId) %>%
  summarise(average_rating = mean(rating))

movies_with_avg_rating <- movies %>%
  left_join(avg_ratings, by = "movieId")
head(movies_with_avg_rating)
```

```
## # A tibble: 6 x 6
##   movieId title                                    genres New_Title  Year average_rating
##     <dbl> <chr>                                    <chr>  <chr>     <dbl>          <dbl>
## 1       1 Toy Story (1995)                         Adven~ "Toy Sto~  1995           3.92
## 2       2 Jumanji (1995)                           Adven~ "Jumanji~  1995           3.43
## 3       3 Grumpier Old Men (1995)          Comed~ "Grumpie~  1995           3.26
## 4       4 Waiting to Exhale (1995)         Comed~ "Waiting~  1995           2.36
## 5       5 Father of the Bride Part II (19~ Comedy "Father ~  1995           3.07
## 6       6 Heat (1995)                      Actio~ "Heat "    1995           3.95
```

```
# plot average ratings by user
average_ratings <- rowMeans(ratings)
ggplot(data.frame(Average_Rating = average_ratings), aes(x = Average_Rating)) +
  geom_histogram(fill = "steelblue", color = "red", bins = 30) +
  ggtitle("Distribution of the Average Rating per User") +
  xlab("Average Rating") +
  ylab("Frequency")
```

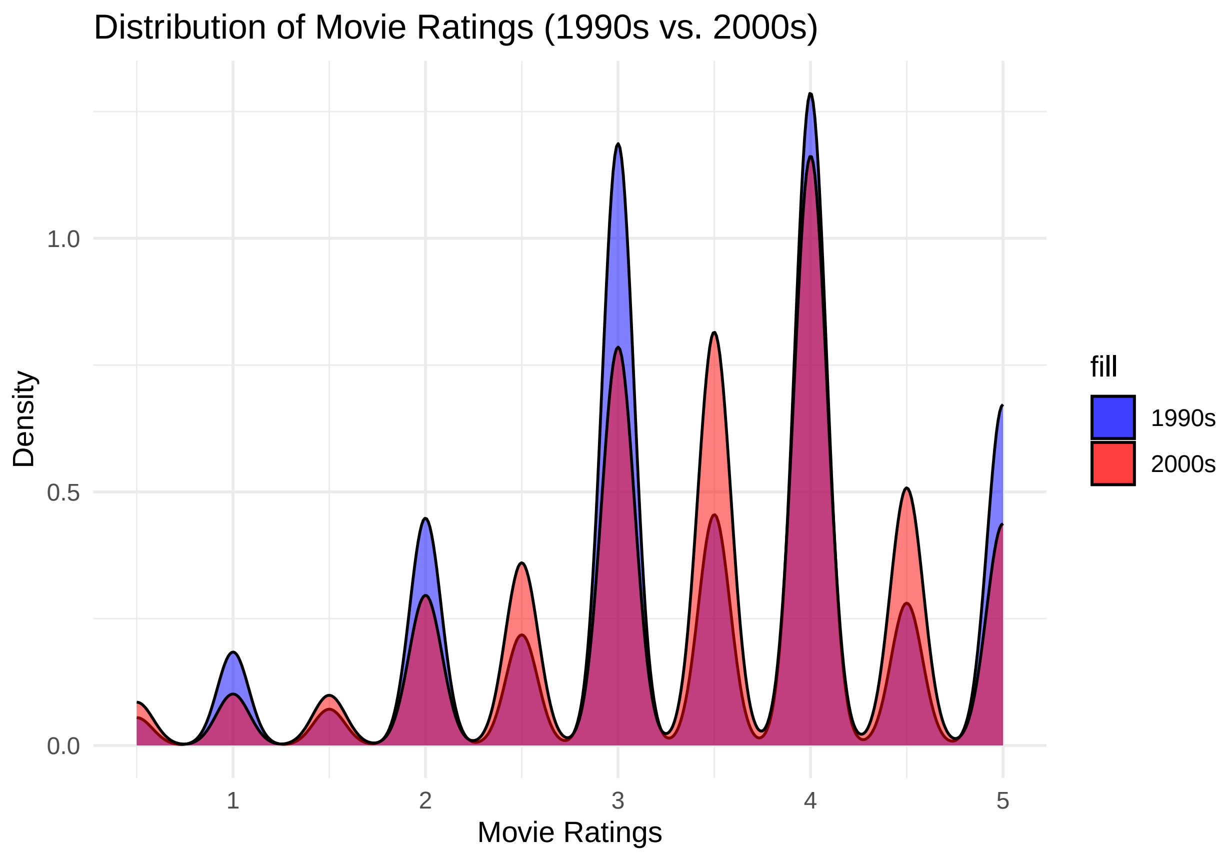


```
library(tidyverse)

correlation_data <- select(merged_df, movieId, genres, Year, rating, timestamp)

correlation_matrix <- cor(correlation_data[, c( "timestamp", "rating")])

print(correlation_matrix)
```

```
##              timestamp      rating
## timestamp  1.00000000 -0.00576634
## rating    -0.00576634  1.00000000
```

```r
# Filter data for movies from the 1990s and 2000s
movies_90s <- merged_df[merged_df$Year >= 1990 & merged_df$Year <= 1999, ]
movies_2000s <- merged_df[merged_df$Year >= 2000 & merged_df$Year <= 2009, ]

# Plot distribution of movie ratings for the 1990s and 2000s
library(ggplot2)

ggplot() +
  geom_density(data = movies_90s, aes(x = rating, fill = "1990s"), alpha = 0.5) +
  geom_density(data = movies_2000s, aes(x = rating, fill = "2000s"), alpha = 0.5) +
  labs(x = "Movie Ratings", y = "Density", title = "Distribution of Movie Ratings (1990s vs. 2000s)") +
  scale_fill_manual(values = c("1990s" = "blue", "2000s" = "red")) +
  theme_minimal()
```

```
## Warning: Removed 4 rows containing non-finite values ('stat_density()').
## Removed 4 rows containing non-finite values ('stat_density()').
```



```r
movie_data <- data.frame(
  MovieID = c(1:1000),
  Genre = c("Action, Adventure", "Drama", "Comedy", "Action, Comedy", "Adventure, Drama")
)

encoded_data <- model.matrix(~ Genre - 1, data = movie_data)
```

```r
encoded_data_df <- as.data.frame(encoded_data)

head(encoded_data_df)
```

```
##   GenreAction, Adventure GenreAction, Comedy GenreAdventure, Drama GenreComedy
## 1                      1                   0                     0           0
## 2                      0                   0                     0           0
## 3                      0                   0                     0           1
## 4                      0                   1                     0           0
## 5                      0                   0                     1           0
## 6                      1                   0                     0           0
##   GenreDrama
## 1          0
## 2          1
## 3          0
## 4          0
## 5          0
## 6          0
```

```r
#Create ratings matrix. Rows = userId, Columns = movieId
ratingmat <- dcast(ratings, userId~movieId, value.var = "rating", na.rm=FALSE)
ratingmat <- as.matrix(ratingmat[,-1]) #remove userIds

#Convert rating matrix into a recommenderlab sparse matrix
ratingmat <- as(ratingmat, "realRatingMatrix")

#Normalize the data
ratingmat_norm <- normalize(ratingmat)

#Create Recommender Model. "UBCF" stands for User-Based Collaborative Filtering
recommender_model <- Recommender(ratingmat_norm, method = "UBCF", param=list(method="Cosine",nn=30))
```

```
## Warning in .local(x, ...): x was already normalized by row!
```

```r
recom <- predict(recommender_model, ratingmat[1], n=10) #Obtain top 10 recommendations for 1st user in
recom_list <- as(recom, "list") #convert recommenderlab object to readable list
na.omit(recom_list)
```

```
## $`0`
##  [1] "3567"  "913"   "55276" "30803" "27611" "4223"  "5066"  "42728" "55052"
## [10] "5882"
```

```r
#Obtain recommendations
recom_result <- matrix(0,10)

for (i in 1:10) {
    recom_result[i] <- merged_df[recom_list[[1]][i], 2]
  }

recom_result
```

```
##       [,1]
##  [1,] "Braveheart (1995)"
##  [2,] "Casino (1995)"
##  [3,] "Where the Money Is (2000)"
##  [4,] "Star Trek IV: The Voyage Home (1986)"
##  [5,] "Dead Poets Society (1989)"
##  [6,] "Apollo 13 (1995)"
##  [7,] "Johnny Mnemonic (1995)"
##  [8,] "Desperately Seeking Susan (1985)"
##  [9,] "Bell, Book and Candle (1958)"
## [10,] "Billy Madison (1995)"
```

Analysis

The data unfolded captivating narratives:

- The distribution of ratings sketched a landscape of viewer sentiments, guiding us toward genres that captured hearts.
- The correlation between timestamp and ratings were also seen, which gave us an understanding that users' ratings are not substantially impacted by the time they watch a movie or provide their feedback.
- There was also a trend of older movies (pre-2000) generally receiving higher ratings, supporting the hypothesis of nostalgia's influence on user preferences.
- I took a sample of the movie dataset to convert genre into multiple binary variables.

Implications for the Consumer

The consumer impact is profound. An effective movie recommendation system has the potential to transform the movie-watching experience. Users will find themselves engrossed in a tailored selection of movies that cater to their preferences. This means discovering hidden gems, avoiding disappointing experiences, and ensuring each movie choice aligns with their tastes.

Limitations

I worked with a relatively small set of attributes and data. Expanding the dataset to include user demographics, contextual information, budget, director, and richer genre information could lead to more accurate recommendations. These variables could reveal how individual movie choices and preferences are influenced by various factors, providing deeper insights into user behavior. Moreover, the model's accuracy could be enhanced through more advanced techniques like matrix factorization or deep learning. Ensuring real-time interactions and feedback are also critical for a robust recommendation system.

Conclusion

Considering the data at hand and the goals of building a movie recommendation system, a hybrid approach that combines both collaborative filtering and content-based filtering methods seems promising. Collaborative filtering leverages the collective behavior of users to suggest movies based on their similarity to other users' preferences. Meanwhile, content-based filtering examines the intrinsic features of movies, like genres, timestamps, and user behaviors, to make recommendations. The next steps would involve refining these techniques, possibly implementing matrix factorization or deep learning algorithms for collaborative filtering and employing natural language processing to extract more information from user reviews for content-based filtering. Additionally, incorporating sentiment analysis to understand user sentiments towards specific genres or movies could enhance the recommendations. The final approach could involve an ensemble of these methods, employing machine learning techniques like logistic regression to combine their outputs effectively. To ensure the system's effectiveness, regular updates and feedback mechanisms should be incorporated, iteratively improving the recommendations and user experience.