HarvardX Data Science Capstone 1: MovieLens Report

Justin Nielson

May 23, 2019

1. Introduction

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

2. Overview

3. Executive Summary - MovieLens edx dataset

Loading the required packages:

Reading in the MovieLens 10M dataset and splitting into edx and validation data sets:

```r
# MovieLens 10M dataset:
# https://grouplens.org/datasets/movielens/10m/
# http://files.grouplens.org/datasets/movielens/ml-10m.zip

# Since I am using using R 3.6.0 I downloaded edx.rds and validation.rds datasets from
# HarvardX_Capstone_MovieLens Google Drive
# https://drive.google.com/drive/folders/1IZcBBX0OmL9wu9AdzMBFUG8GoPbGQ38D

movielens <- readRDS("edx.rds", refhook = NULL)
validation <- readRDS("validation.rds", refhook = NULL)

# Validation set will be 10% of MovieLens data
# if using R 3.6.0: set.seed(1, sample.kind = "Rounding")
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.1, list = FALSE)
edx <- movielens[-test_index,]
temp <- movielens[test_index,]

# Make sure userId and movieId in validation set are also in edx set

validation <- temp %>%
  semi_join(edx, by = "movieId") %>%
  semi_join(edx, by = "userId")

# Add rows removed from validation set back into edx set

removed <- anti_join(temp, validation)
edx <- rbind(edx, removed)

rm(dl, ratings, movies, test_index, temp, movielens, removed)
```

4. Methods and Analysis:

4.1. Data exploration and visualization

```
# Data exploration of MovieLens edx and validation datasets

edx <- data.frame(edx)
edx$timestamp <- as_datetime(edx$timestamp)
glimpse(edx)
```

```
## Observations: 8,100,065
## Variables: 6
## $ userId    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ movieId   <dbl> 122, 292, 316, 329, 355, 356, 362, 364, 370, 377, 42...
## $ rating    <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5...
## $ timestamp <dttm> 1996-08-02 11:24:06, 1996-08-02 10:57:01, 1996-08-0...
## $ title     <chr> "Boomerang (1992)", "Outbreak (1995)", "Stargate (19...
## $ genres    <chr> "Comedy|Romance", "Action|Drama|Sci-Fi|Thriller", "A...
```

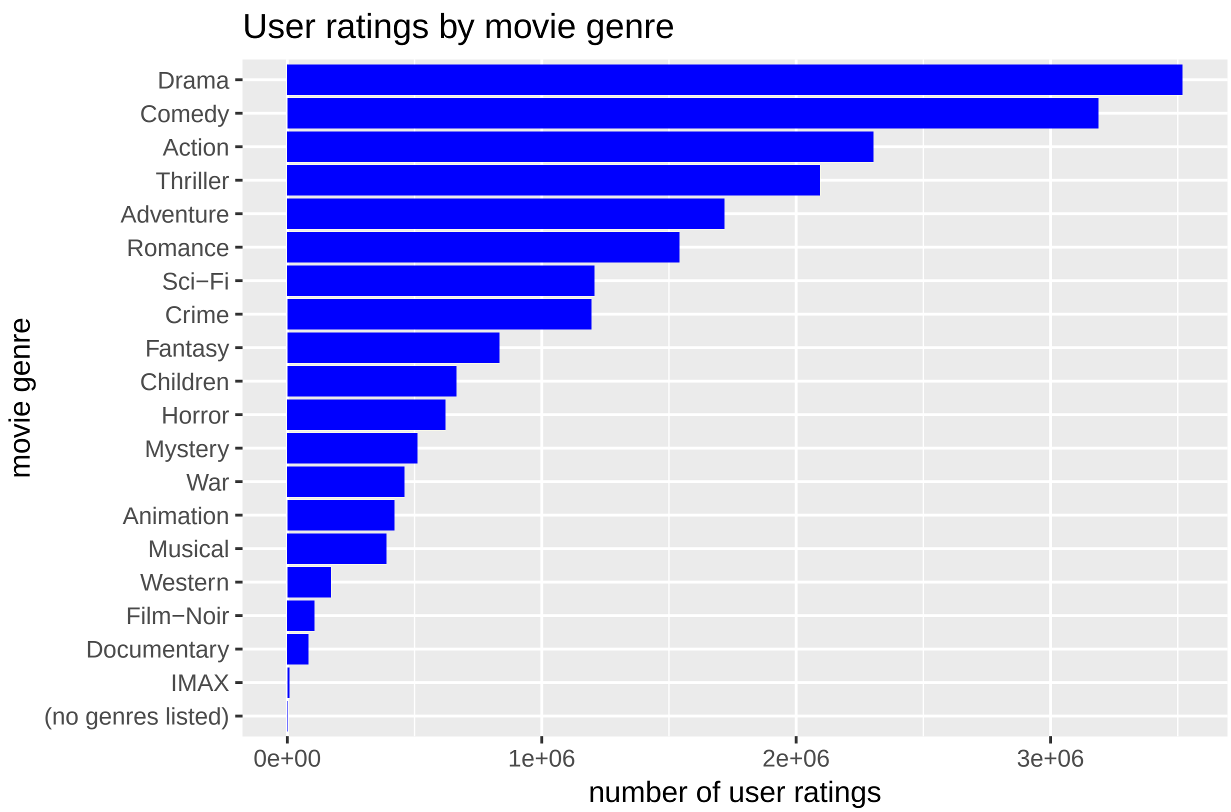
```
validation <- data.frame(validation)
validation$timestamp <- as_datetime(validation$timestamp)
glimpse(validation)
```

```
## Observations: 899,990
## Variables: 6
## $ userId    <int> 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 5, 5, 5, 5, 5, 5, 5...
## $ movieId   <dbl> 185, 260, 590, 1049, 1210, 1148, 1552, 3684, 6539, 4...
## $ rating    <dbl> 5.0, 5.0, 5.0, 3.0, 4.0, 4.0, 2.0, 4.5, 5.0, 3.0, 3....
## $ timestamp <dttm> 1996-08-02 10:58:45, 1997-07-07 03:02:42, 1997-07-0...
## $ title     <chr> "Net, The (1995)", "Star Wars: Episode IV - A New Ho...
## $ genres    <chr> "Action|Crime|Thriller", "Action|Adventure|Sci-Fi", ...
```

```
# Table of user ratings by movie genre
movie_genre <- edx %>% separate_rows(genres, sep = "\\|") %>%
  group_by(genres) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

movie_genre <- data.table(movie_genre)
movie_genre <- movie_genre[order(-count),]

ggplot(data=movie_genre, aes(x=reorder(movie_genre$genres,movie_genre$count),y=movie_genre$count,fill=I
  geom_bar(position="dodge",stat="identity") +
  coord_flip() +
  labs(x="movie genre", y="number of user ratings", caption = "source data: MovieLens edx dataset") +
  ggtitle("User ratings by movie genre")
```

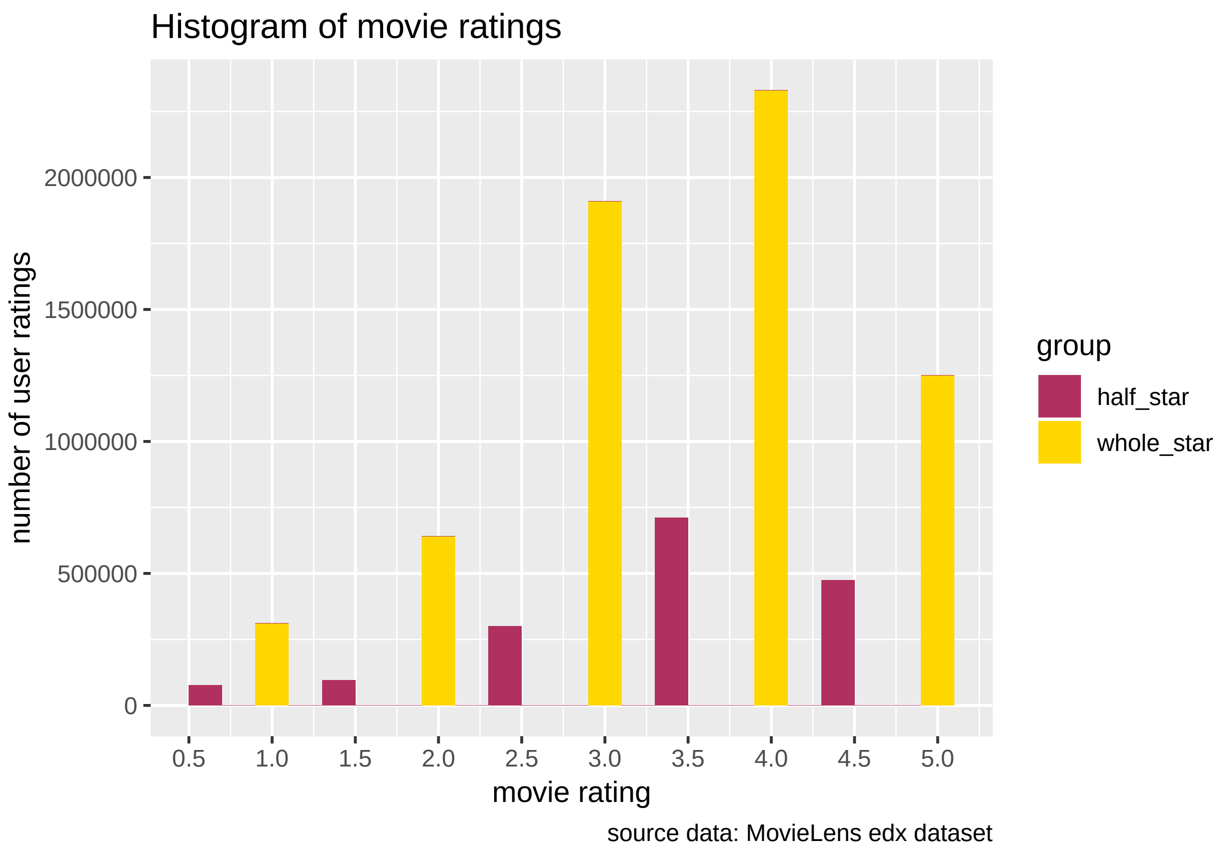


source data: MovieLens edx dataset

```
# Histogram of MovieLens edx ratings
group <-  ifelse((edx$rating == 1 |edx$rating == 2 | edx$rating == 3 |
                    edx$rating == 4 | edx$rating == 5) ,
                 "whole_star",
                 "half_star")

edx_ratings <- data.frame(edx$rating, group)

ggplot(edx_ratings, aes(x= edx$rating, fill = group)) +
  geom_histogram( binwidth = 0.2) +
  scale_x_continuous(breaks=seq(0, 5, by= 0.5)) +
  scale_fill_manual(values = c("half_star"="maroon", "whole_star"="gold")) +
  labs(x="movie rating", y="number of user ratings", caption = "source data: MovieLens edx dataset") +
  ggtitle("Histogram of movie ratings")
```



Histogram of movie ratings

source data: MovieLens edx dataset

4.2. Data preprocessing and transformation

4.3. Evaluated Machine Learning Algorithms

4.3.1 Logistic Regression

4.3.2 K-Nearest Neighbors (K-NN)

4.3.3 Support Vector Machine (SVM)

4.3.4 Kernel SVM

4.3.5 Naive Bayes

4.4.6 Decision Tree

4.4.7 Random Forest

4.4.8 Ensemble Method

5. Results:

6. Conclusion:

References

+ Adhikari, A. and DeNero, J., 2019. Computational and Inferential Thinking,

https://www.inferentialthinking.com/chapters/intro.html

+ Eremenko, K. and de Ponteves, H., 2019. Machine Learning A-Z™: Hands-On Python & R In #### Data Science, https://www.udemy.com/machinelearning/

+ Irizzary,R., 2018. Introduction to Data Science,

github page,https://rafalab.github.io/dsbook/

+ Koren, Y., 2009. The BellKor Solution to the Netflix Grand Prize.

Netflix prize documentation,

https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf