Individual Assignment Tan Bui BD2

Inhoud

Summary

For the individual assignment we had to come up with a research question. Since we had to use MovieLens data and another external source for data. I came up with the following,

What happens to the movie ratings over the years? Do ratings go up as years go by? Or are they rather dropping? Guess we'll find out.

I use mongoDB to store the movielens data which was a must requirement by HvA.

Also I used www.Mockaroo.com to create dummy data to compare the movielens data with. It won't be as realistic as actual data but for an assignment like this I can still learn how to use R to my advantage while having lots of data. I also got permission from my lecturer to make use of this.

I use the following libraries in R

```
#install.packages("shinydashboard")
#install.packages("DT")
#install.packages("ggplot2")
#install.packages("RMongo")
#install.packages("dplyr")
#install.packages("sqldf")
#install.packages("plotly")
#install.packages("tidyr")
#install.packages("stringr")
#install.packages("stringi")
```

Research question

What happens to the movie ratings over the years? Do ratings go up as years go by? Or are they rather dropping? Guess we'll find out.

Data discovery

Movielens

I got the movielens data from https://grouplens.org/datasets/movielens/ which was required for this assignment. There were several datasets which could be used. And I've tried out a few of them, the older data sets were more complicated in my opinion. So at last, I took ml-latest-small as a dataset because it looked like a good start to get into data analysis.

Mockaroo

Also I used www.Mockaroo.com to create dummy data to compare the movielens data with. It won't be as realistic as actual data but for an assignment like this I can still learn how to use R to my advantage while having lots of data. I also got permission from my lecturer to make use of this.

Data preparations

How are the datasets stored?

One dataset had to run live with a database connection (SQL or NOSQL).

I chose MongoDB which is a NOSQL database. Although I prefer SQL over NOSQL, I use MongoDB with the knowledge that you can store (almost) any data within MongoDB without having to worry.

This is why I just dumped all Movielens data in MongoDB and have R make a connection to retrieve it.

Retrieving the data

Code to retrieve the data

```
mockdata <- read.csv("C:/Users/Tan/Desktop/Big data/Data processing/MOCK_DATA2.csv")

mcon <- mongoDbConnect("Movies", port=27017)

mlMovies <- RMongo::dbGetQuery(mcon, "ml-movies","{}", skip=0, limit=999999)
mlRating <- RMongo::dbGetQuery(mcon, "ml-ratings","{}", skip=0, limit=999999)
mlTags <- RMongo::dbGetQuery(mcon, "ml-tags","{}", skip=0, limit=999999)
```

Combining the data

I used the library sqldf to be able to do sql queries within data frames. Also joining 2 movielens dataframes together.

Then I got the movielens data ratings grouped by year with the average rating in that year stored in a dataset named “ml”

I did the similar thing to mockaroo data which is the mockdata and named “md”

```
mlMoviesRatings <- sqldf("SELECT movieID, title, rating, year FROM mlMovies JOIN mlRating USING(movieID)")

ml <- mlMoviesRatings %>%
  group_by(year) %>%
  summarise(
    avg_rating = mean(rating),
    dataset = "ml"
  )

md <- mockdata %>%
  group_by(movie_date) %>%
  summarise(
    avg_rating = mean(movie_rating),
    dataset = "md"
  )
colnames(md) <- colnames(ml)
```

What kind of processing was needed?

The data had to be manipulated to get exactly what I wanted.

Before you had a pattern ”Toystory (1995)” in the title and I wanted to create a seperate column for 1995 which will be named year.

I did the following.

```
#Getting the year from the title with regex: \\([0-9]{4}\\)
year <- data.frame(year = sapply(mlMovies$title, function(x) str_extract(x, "\\([0-9]{4}\\)")))
#Removing the year from the title
mlMovies$title <- gsub("\\([0-9]{4}\\)", "", mlMovies$title)
#Adding year to movielens in separate column
mlMovies$year <- as.numeric(sapply(year, function(x) stri_sub(str = x, from = 2, to = 5)))
```

str_extract is a function which extracts the pattern found which in this case is the year.

Gsub removes 4 numbers after the title in this case.

At last add a column named year with the values that have been extracted.

Data visuals

Shinydashboard

Plotly

DT (datatable)

Ggplot2

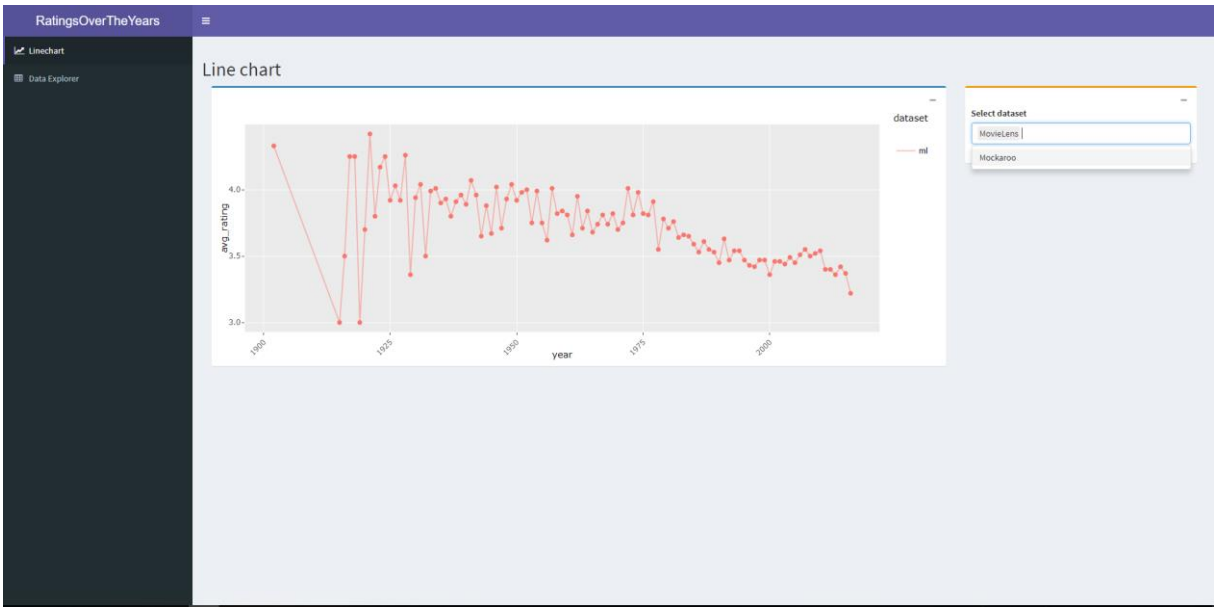
I used these libraries to create the results

```
  status = "warning",
  selectInput("dataset1", "Select dataset", c("MovieLens" = "ml", "Mockaroo" = "md"), multiple = TRUE, selected = "ml
)

function(input, output, session) {

  #Creating reactive object: result
  result <- reactive({
      filter(ml, dataset %in% input$dataset1)
  })

  #Rendering Line chart
  output$plot1 <- renderPlotly({

    p <-
      ggplot(data = result(),
             aes(x = year,
                 y = avg_rating,
                 color = dataset)) +
      geom_line(alpha = 0.5) +
      geom_point(aes(group = dataset)) +
      theme(axis.text.x = element_text(
        angle = 45,
        hjust = 1,
        vjust = 0.5
      ))

    p <- ggplotly(p)
  })
```

By putting the moviedata as options to select and creating a reactive object I can make the linechart somewhat interactive. As followed by rendering the line itself.

Results


127.0.0.1:4347
Apps
Enabling the Google
RatingsOverTheYears
Linechart
Data Explorer
Line chart
dataset
md
avg_rating
year
1940
1960
1980
2000
Select dataset
Mockaroo
MovieLens


RatingsOverTheYears
Linechart
Data Explorer
Line chart
dataset
ml
avg_rating
year
1900
1925
1950
1975
2000
Select dataset
MovieLens
Mockaroo

Here are the linegraphs which shows the avg_rating in the y-axis and year on the x-axis.

It is kinda interactive because you can switch up which dataset is shown by selecting.

Conclusion

What happens to the movie ratings over the years? Do ratings go up as years go by? Or are they rather dropping? Guess we'll find out.

Answer,

Movies that came out recent years are more consistent in ratings than older ones. Although they do not go up or drop by much in recent years. That according to the movielens data.

The reason why can't be said, and that might be the key in big data. You can't make statements with limited data.

The mockaroo dataset consist of dummy data but was fun experimenting with.