Northeastern University, Khoury College of Computer Science

CS 6220 Data Mining — Assignment 1

Due: January 18, 2023(100 points)

Wanqing Wang
wqwang-cerealk
wang.wa@northeastern.edu

1 Getting Started - 10 points

1.1 Using Docker

Different companies use different tools for development and different work environments. For future assignments, we won't be prescriptive, but in this homework, we're going to familiarize ourselves with some of the most useful and common delivery and development environment tools in industry today.
Docker http://www.Docker.com is a useful mechanism for delivering software or scaling it up. For example, say we want to run a multi-computer job, passing *Docker containers* to each of the nodes in the cluster is one way to have repetitive and predictable behavior when doing large scale compute.
There are two essential Docker units: a **container** and a **container image**.

1. A **container** is a sandboxed process on your machine that is isolated from all other processes on the host machine. That isolation leverages kernel namespaces and cgroups, features that have been in Linux for a long time. Docker has worked to make these capabilities approachable and easy to use. To summarize, a container:
   a) is a runnable instance of an image. You can create, start, stop, move, or delete a container using the DockerAPI or CLI.
   b) can be run on local machines, virtual machines or deployed to the cloud.
   c) is portable (can be run on any OS).
   d) is isolated from other containers and runs its own software, binaries, and configurations.
2. When running a container, it uses an isolated filesystem. This custom filesystem is provided by a **container image**. Since the image contains the container's filesystem, it

> must contain everything needed to run an application - all dependencies, configurations, scripts, binaries, etc. The image also contains other configuration for the container, such as environment variables, a default command to run, and other metadata.

Go ahead and download and install Docker. The getting started guide on Docker has detailed instructions for setting up Docker on

- Mac https://docs.docker.com/desktop/install/mac-install/,
- Linux https://docs.docker.com/install/linux/docker-ce/ubuntu
- Windows https://docs.docker.com/docker-for-windows/install.

1.2 Executing Your "Hello World"

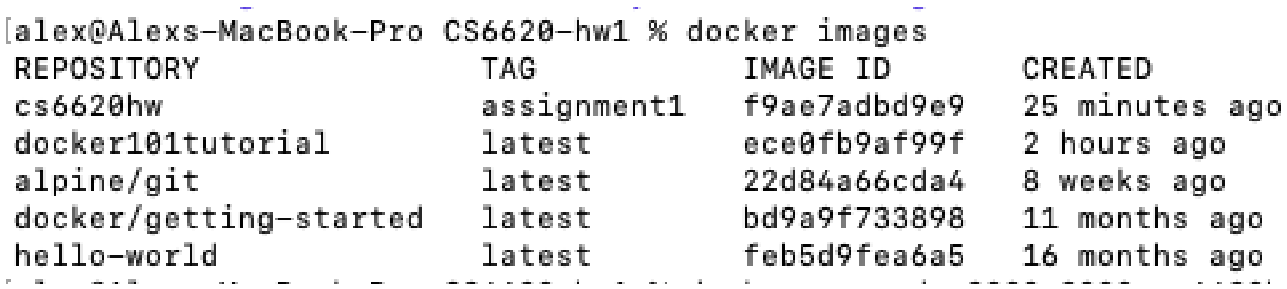
For this assignment, we'll start with creating a Dockerfile in your submission folder. Specify the operating system and version of Python in the Dockerfile. You will subsequently need to install Python and libraries that you anticipate importing. Do not add the data into the image; you will need to pass that into the container with the `-v` Docker option.
For example, here's the most basic `Dockerfile`:

```
FROM ubuntu:20.04

RUN apt update && apt install -y sbcl
WORKDIR /usr/src
```

- The assignment1.py file is only for Section 1 Hello World in Docker, the rest of the codes are in assignment1.ipynb file

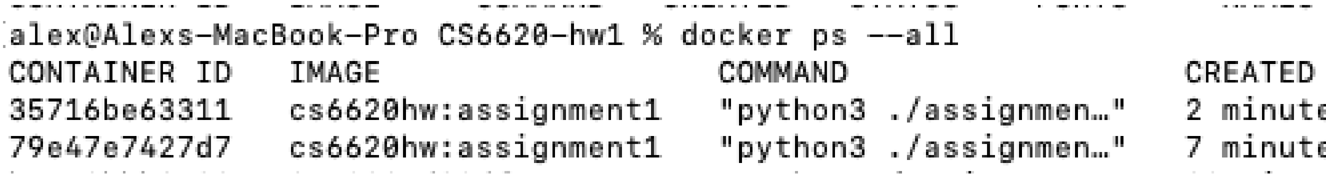
For this assignment, you'll set up your Docker environment and the appropriate versions of Python. Specifically,

1. Download and install Docker
2. Create your Dockerfile
   - //Command for create a docker file
   - touch Dockerfile
3. Compile your Docker image
   - //Command for compiling Docker image
   - docker image build -t cs6220hw1:assignment1 /Users/alex/desktop/CS6620-hw1
4. Screenshot a list of the Docker images available
   - //Command for checking images
   - docker images



```
[alex@Alexs-MacBook-Pro CS6620-hw1 % docker images
REPOSITORY                TAG           IMAGE ID       CREATED
cs6620hw                  assignment1   f9ae7adbd9e9   25 minutes ago
docker101tutorial         latest        ece0fb9af99f   2 hours ago
alpine/git                latest        22d84a66cda4   8 weeks ago
docker/getting-started    latest        bd9a9f733898   11 months ago
hello-world               latest        feb5d9fea6a5   16 months ago
```

5. Screenshot a list of the running Docker containers that include one with the image you created (Since it's hard to capture the moment Hello World runs, I just list all containers)
   - //Command for run image in containers and check all containers
   - docker run cs6620hw:assignment1
   - docker ps –all



```
alex@Alexs-MacBook-Pro CS6620-hw1 % docker ps --all
CONTAINER ID   IMAGE                   COMMAND                   CREATED
35716be63311   cs6620hw:assignment1    "python3 ./assignmen…"    2 minute
79e47e7427d7   cs6620hw:assignment1    "python3 ./assignmen…"    7 minute
```

6. Include both screenshots and the command you used in your write up

1.3 Github

Software version control at companies is essential for every software company in the industry. There are several types, including *Subversion/SVN* (which Google uses its in-house version branched from SVN). The most popular tool of choice is Github, which Microsoft recently bought.
At the end of this assignment, your submission will point to a repository, where the following files will be reviewed and subsequently graded:

- `Dockerfile` specifying what packages that you've used
- `assignment1.tex` file with your homework writeup
- `assignment1.pdf` file of the compiled version of your *.tex file
- `assignment1.py` file of your working code

None of the other files in that repository will be reviewed. We've provided a LaTeXtemplate that you can use for submission, provided here:

- https://github.com/kni-neu/homework-1/blob/main/assignment1-questions.tex

Do *NOT* include data into your Git repository. If you need help with LaTeX, the program that creates a PDF file from a coded text file (with extension `*.tex`), you may wish to use the online site `overleaf.com`. There is a helpful guide at this url:
https://www.overleaf.com/learn

2 Identifying All Sets - 40 points

In subsequent lectures, you'll learn about frequent item sets, where relationships between items are learned by observing how often they co-occur in a set of data. This information is useful for making recommendations in a rule based manner. Before looking at frequent item sets, it is worth understanding the space of all possible sets and get a sense for how quickly the number of sets with unique items grows.
Suppose that we've received only a hundred records of items bought by customers at a market. Each line in the file represents the items an individual customer bought, i.e. their basket. For example, consider the following rows.

- Please refer the .ipynb file Section 2 for functions and codes for Section 2

```
ham, cheese, bread
dates, bananas
celery, chocolate bars
```

Customer 1 has a basket of ham, cheese, and bread. Customer 2 has a basket of dates and bananas. Customer 3 has a basket of celery and chocolate bars. Each of these records is the receipt of a given customer, identifying what they bought.

1. What is the cardinality of the full set of unique items? Write a function called `cardinality_items` that takes a `.csv` text string file as input, where the format is as the above, and calculates the cardinality of the dataset.

2. Taking any `.csv` file as a sample of a larger dataset, we'd occasionally like to understand the space of all possible subsets comprised of unique items. If there are N unique items (i.e., the cardinality of the entire dataset is N), how many sets with unique items can there possibly be? (Ignore the null set.) NOTE: I only expect the formula, and there is no code associated with this question.

   a) We want to compute the number of unique subsets so the total number is (number of subsets which has size of 1) + (number of subsets which has size of 2) + ... + (number of subsets which has size of N)

   b) The formula should be ${}_NC^1 + {}_NC^2 + ... + {}_NC^N = 2^N - 1$ since we want to ignore the empty subset

3. Write a module called `all_itemsets()` with the following input/output:

   a) Input: $filename$ = the `.csv` text string file, where the format is as the above.

   b) Output: $L = [S_1, S_2, \cdots S_M]$, which is a list of all possible sets of with unique items N

4. Let's take the small sample `.csv` provided as reflective of the distribution of the receipts writ large. So, for example, if the set $S = \{\text{bread}, \text{oatmeal}\}$ occurs twice in a dataset with 100 records, then the probability of item set $\{\text{bread}, \text{oatmeal}\}$ occurring is 0.02. Write a module called `prob_S` with the following input/output:

   a) Input:
   S = the set in question
   D = the entire Dataset (which if it's in memory, Python will pass by reference). In this case, D can be a list of lists or a list of sets:

- `[ [A, B], [A, C], [C, D] , ... ]`
- `[ {A, B}, {A, C}, {C, D} , ... ]`

b) Output: $P(S)$ = the probability that S occurs

3 The Netflix Challenge - 50 points

One of the most famous challenges in data science and machine learning is Netflix's Grand Prize Challenge, where Netflix held an open competition for the best algorithm to predict user ratings for films. The grand prize was $1,000,000 and was won by BellKor's Pragmatic Chaos team. This is the dataset that was used in that competition.

- https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data

In this exercise, we're going to do a bit of exploring in the Netflix Data. Start by downloading the data. If all worked out well, you should have the files in Fig. 3.1. The Kaggle dataset is close to 700MB large, and may take a long time to download. Do *not* include this data in your Docker container, but rather, mount the folder with the data.

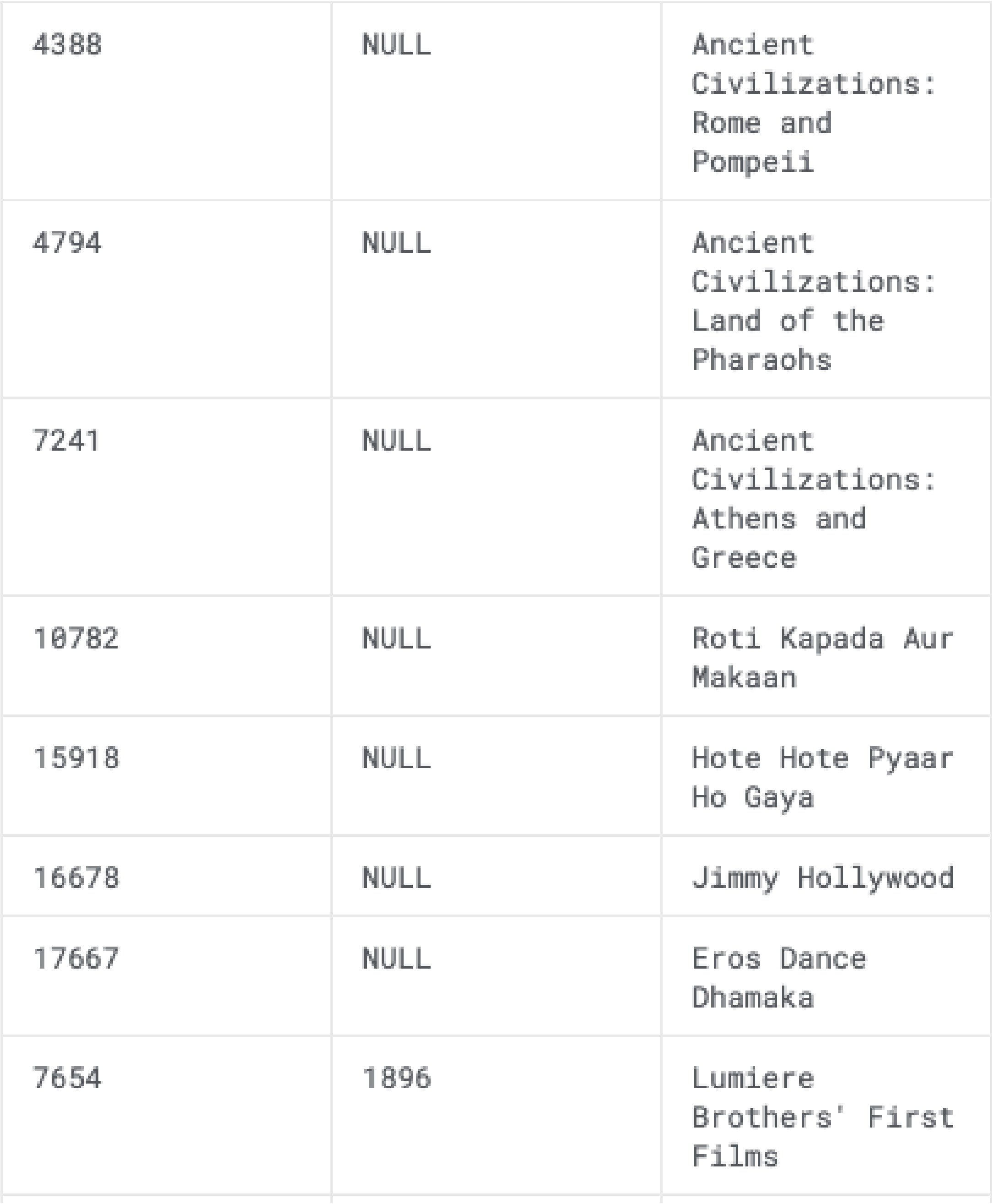
- Please refer the .ipynb file Section 3 for functions and codes for Section 3

3.1 Data Verification

Data integrity tends to be a problem in large scale processing, especially if there is little to no support. Therefore, it's important to verify the quality of the file download.

1. A large part of machine learning and data science is about getting data in the right format. Verify that the schema is the same as the Kaggle Dataset's description. Add screenshots to your assignment.

   a) The movie title dataset looks quite similar to Kaggle's description. However, I noticed there are seven movies do not have release date as the description says.



4388	NULL	Ancient Civilizations: Rome and Pompeii
4794	NULL	Ancient Civilizations: Land of the Pharaohs
7241	NULL	Ancient Civilizations: Athens and Greece
10782	NULL	Roti Kapada Aur Makaan
15918	NULL	Hote Hote Pyaar Ho Gaya
16678	NULL	Jimmy Hollywood
17667	NULL	Eros Dance Dhamaka
7654	1896	Lumiere Brothers' First Films

b) The training dataset looks similar to the description which has 17770 files. And there are three columns for a typical rating record

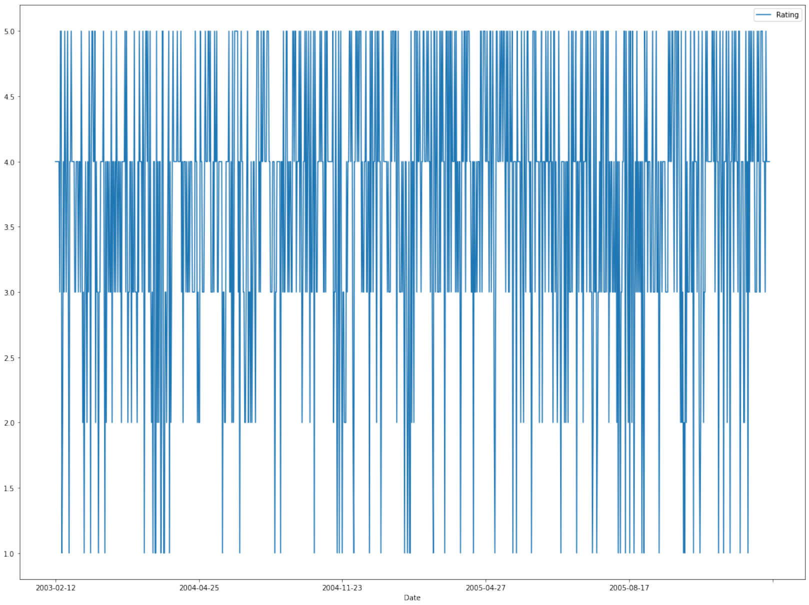
c) The probe and qualifying dataset also seem same to the description since there are two cols in qualifying dataset without empty lines and one col in probe dataset.

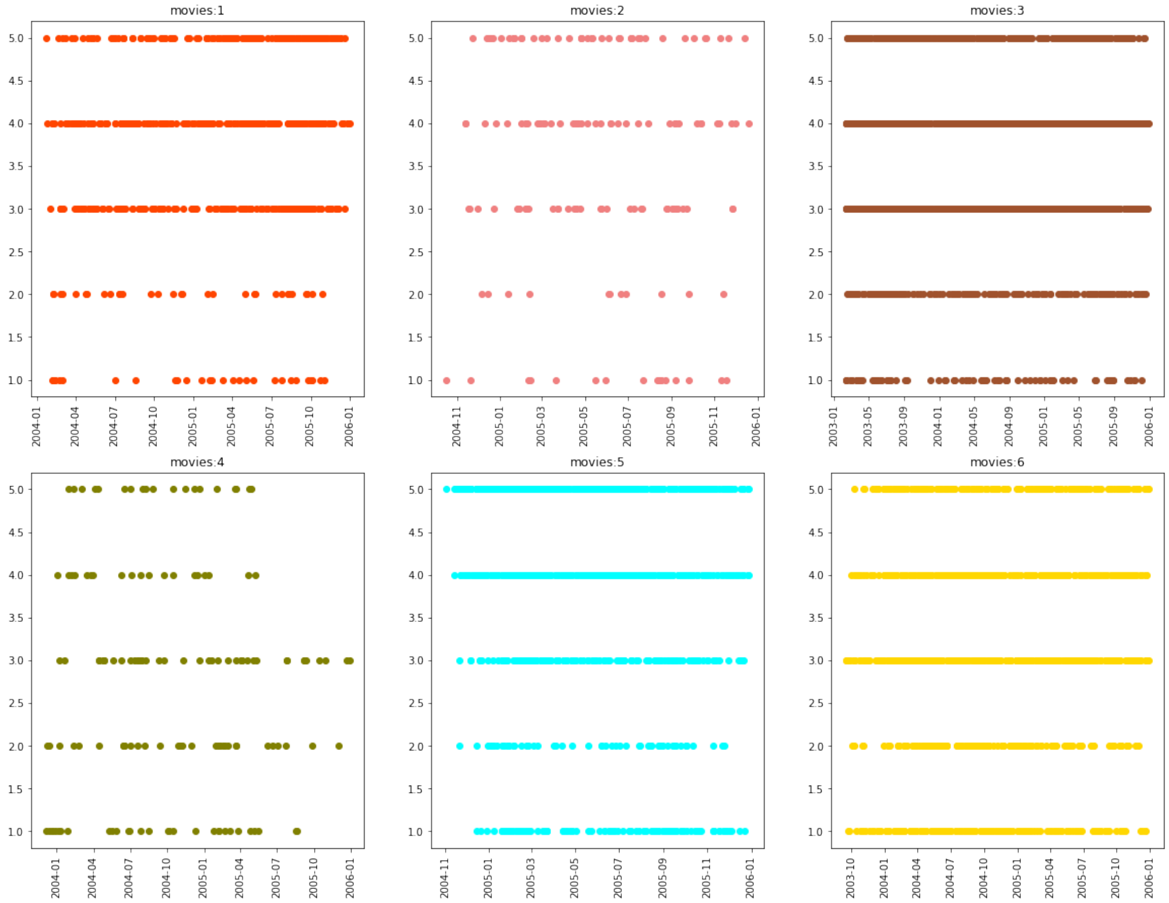
```
file: movie_title.csv, There are 17770 data items, a total of 3 cols

file: combined_data_1.txt, There are 24058263 data items, a total of 3 cols

file: combined_data_2.txt, There are 26982302 data items, a total of 3 cols

file: combined_data_3.txt, There are 22605786 data items, a total of 3 cols

file: combined_data_4.txt, There are 26851926 data items, a total of 3 cols

file: qualifying.txt, There are 2834601 data items, a total of 2 cols

file: probe.txt, There are 1425333 data items, a total of 1 cols
```

3.2 Data Analysis

Let's answer the following questions in your writeup:

1. How many total records are there?
   - There are total 17770 records for movie title dataset, 2817131 records for qualifying dataset, 1408395 records for probe dataset and 100480507 records for training dataset
2. Can you plot the distribution of star ratings over users and time? The granularity of the sliding window is at your discretion. Are there any trends?





   - Based on the graph plotted, overall, I think the ratings over time change is pretty stable. However, some movies show that their ratings become better over time.

3. What percentage of the films have gotten *more* popular over time?

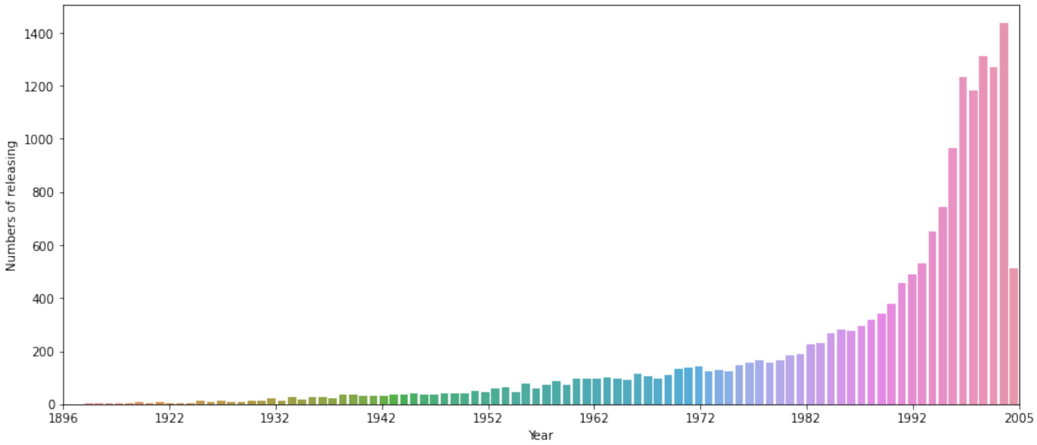
   - Sort each comment based on rating date, count first half's average rating and second half's average rating then compare. If the second value is larger, it indicates that the movie become more popular over time. About 40 percentage of films get more popular over time.

4. How many films have been re-released? How do you know?

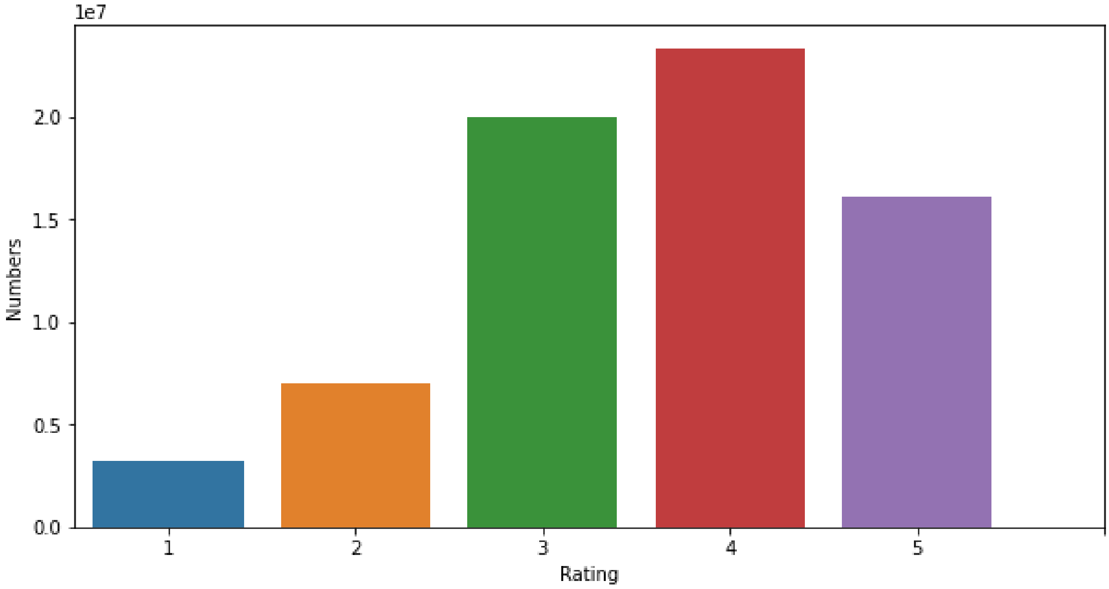
   - I use a sample from part of the training dataset, in combined data 1 dataset, there are 21 movies seem to be re-filmed since the comment data is earlier than the release date. Based on the percentage, I infer that there might be around 80 movies have been re-filmed.

5. What other information might we try to extract to better understand the data? For the questions that you may come up with (especially any time series data), make sure you back up your assertions with plots. Go ahead and play around with the data, and explore.

   - We can see that the number of movie releasing is getting larger over time.



- We can see that most users tend to rates 4 for a movie

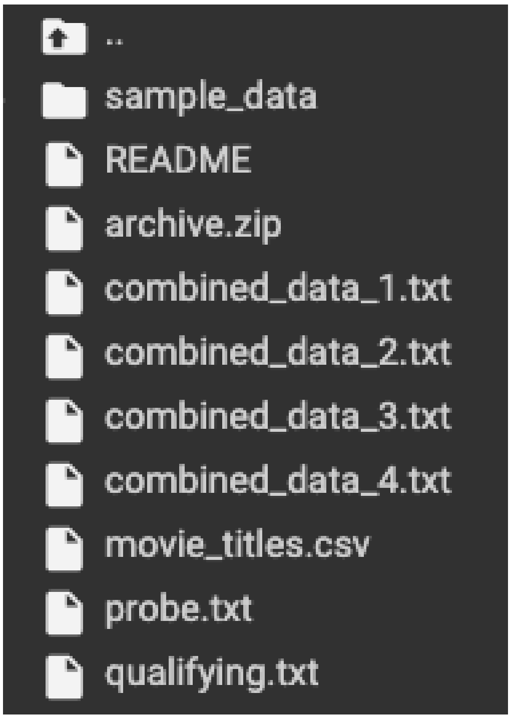


6. What are some interesting problems that we might solve? (No need to actually solve them!)
   - Filter out rarely rated movies and users who don't give enough ratings
   - Get the number of ratings on each day of the week to see do they have any relationship
   - Finding Global Average of all movie ratings, Average ratings per user, and Average rating per movie
   - Use machine learning algorithms to predict film rankings and compare the performance of different algorithms

4 Grading Criterion

A significant portion of the grading rubric is the presentation of your report. We'll review:

1. the answers to questions.
2. your code and its legibility



..
sample_data
README
archive.zip
combined_data_1.txt
combined_data_2.txt
combined_data_3.txt
combined_data_4.txt
movie_titles.csv
probe.txt
qualifying.txt

Figure 3.1: If everything worked out well, you should have the above files available to browse and process.

3. the clarity of your write-up, including
   a) pipeline and code decisions,
   b) perspectives on the solution,
   c) and algorithmic rationale.