Northeastern University, Khoury College of Computer Science

CS 6220 Data Mining — Assignment 2

Due: January 25, 2023(100 points)

YOUR NAME

YOUR GIT USERNAME

YOUR EMAIL

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 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 pro- vided 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

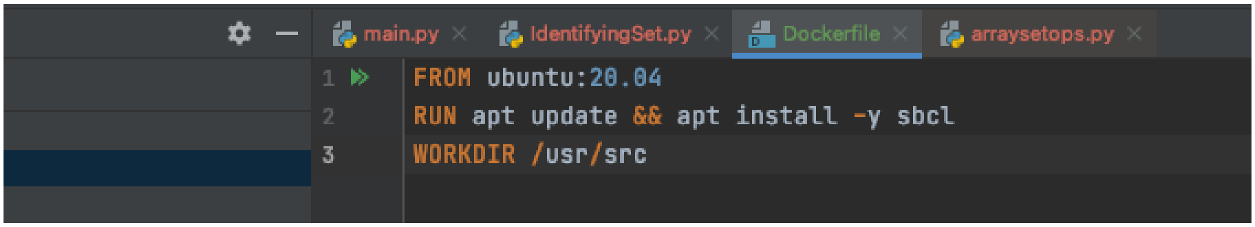
For this assignment, you’ll set up your Docker environment and the appropriate versions of Python. Specifically,

1. Download and install Docker

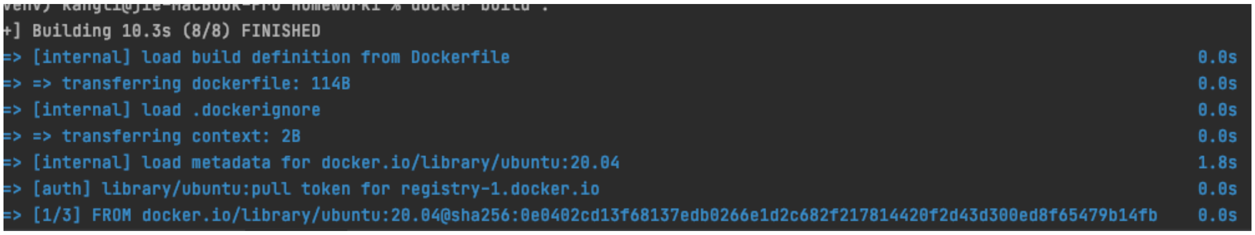
Download from the website and install, write the code in pycharm

2. Create your Dockerfile

Create a dockerfile in the terminal or pycharm as below picture



```
FROM ubuntu:20.04
RUN apt update && apt install -y sbcl
WORKDIR /usr/src
```



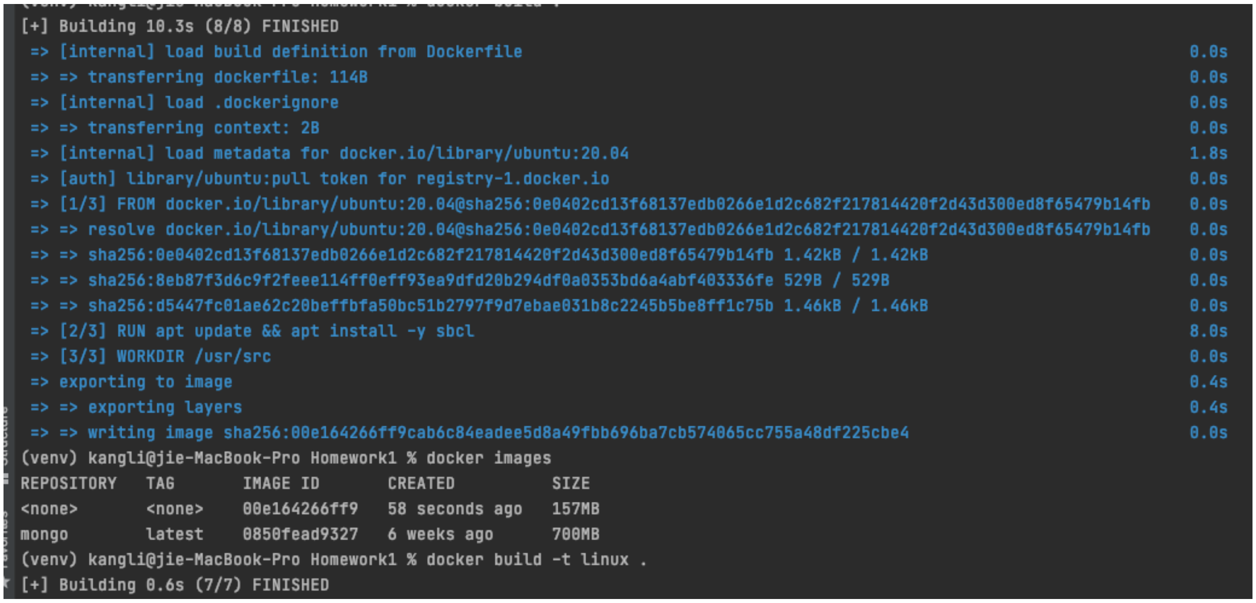
```
venv) kangli@jie-MacBook-Pro homework1 % docker build .
+] Building 10.3s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                                0.0s
=> => transferring dockerfile: 114B                                                                0.0s
=> [internal] load .dockerignore                                                                   0.0s
=> => transferring context: 2B                                                                     0.0s
=> [internal] load metadata for docker.io/library/ubuntu:20.04                                     1.8s
=> [auth] library/ubuntu:pull token for registry-1.docker.io                                       0.0s
=> [1/3] FROM docker.io/library/ubuntu:20.04@sha256:0e0402cd13f68137edb0266e1d2c682f217814420f2d43d300ed8f65479b14fb   0.0s
```

3. Compile your Docker image

Check the default docker image, found two default images, one is mongo, one is <none>

4. Screenshot a list of the Docker images available

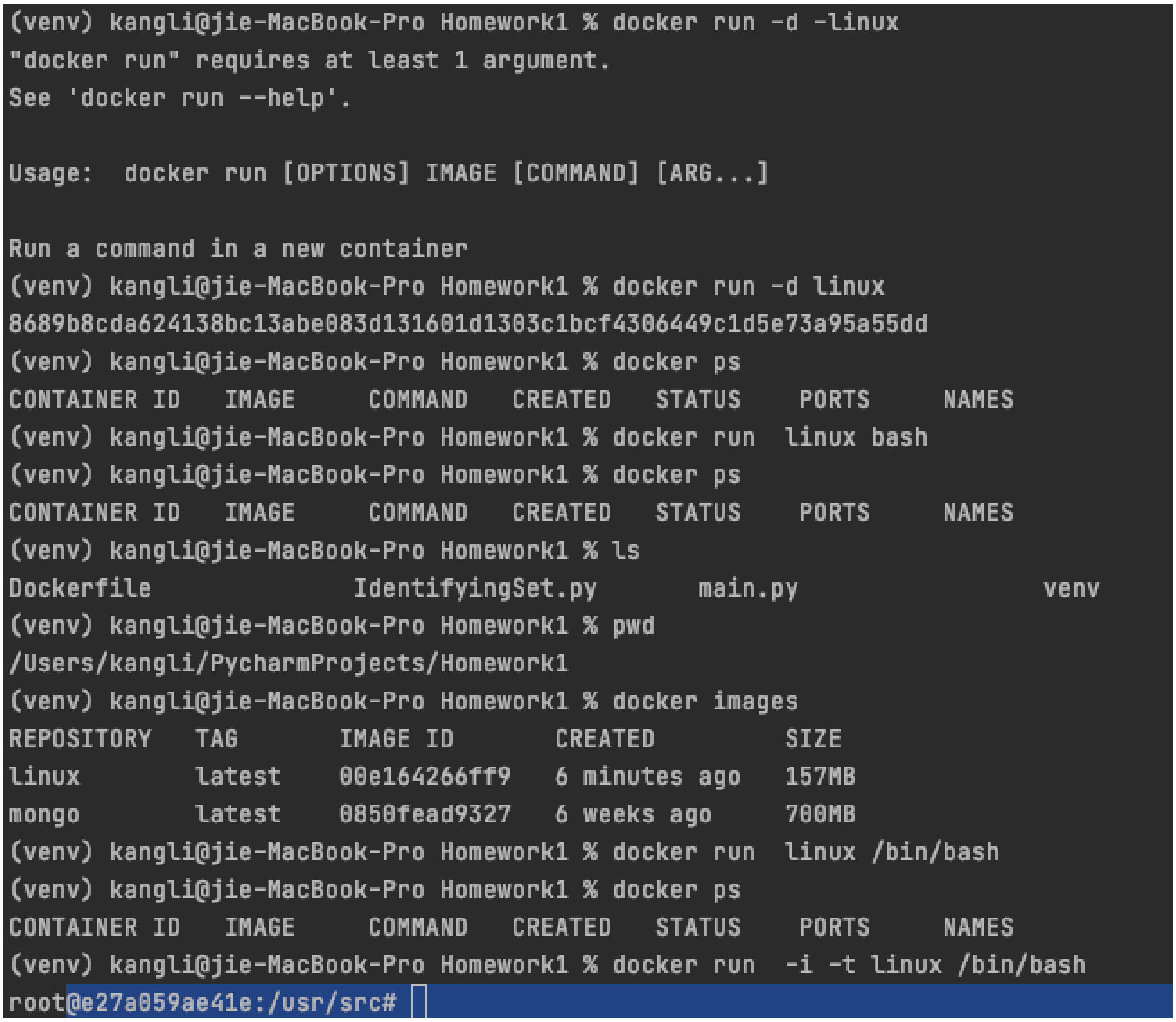
Then I create a docker images named"linux" and run it, check it.



```
[+] Building 10.3s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                                                                     0.0s
 => => transferring dockerfile: 114B                                                                                     0.0s
 => [internal] load .dockerignore                                                                                        0.0s
 => => transferring context: 2B                                                                                          0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                                          1.8s
 => [auth] library/ubuntu:pull token for registry-1.docker.io                                                            0.0s
 => [1/3] FROM docker.io/library/ubuntu:20.04@sha256:0e0402cd13f68137edb0266e1d2c682f217814420f2d43d300ed8f65479b14fb    0.0s
 => => resolve docker.io/library/ubuntu:20.04@sha256:0e0402cd13f68137edb0266e1d2c682f217814420f2d43d300ed8f65479b14fb    0.0s
 => => sha256:0e0402cd13f68137edb0266e1d2c682f217814420f2d43d300ed8f65479b14fb 1.42kB / 1.42kB                           0.0s
 => => sha256:8eb87f3d6c9f2feee114ff0eff93ea9dfd20b294df0a0353bd6a4abf403336fe 529B / 529B                               0.0s
 => => sha256:d5447fc01ae62c20beffbfa50bc51b2797f9d7ebae031b8c2245b5be8ff1c75b 1.46kB / 1.46kB                           0.0s
 => [2/3] RUN apt update && apt install -y sbcl                                                                          8.0s
 => [3/3] WORKDIR /usr/src                                                                                               0.0s
 => exporting to image                                                                                                   0.4s
 => => exporting layers                                                                                                  0.4s
 => => writing image sha256:00e164266ff9cab6c84eadee5d8a49fbb696ba7cb574065cc755a48df225cbe4                             0.0s
(venv) kangli@jie-MacBook-Pro Homework1 % docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
<none>       <none>    00e164266ff9   58 seconds ago   157MB
mongo        latest    0850fead9327   6 weeks ago      700MB
(venv) kangli@jie-MacBook-Pro Homework1 % docker build -t linux .
[+] Building 0.6s (7/7) FINISHED
```

5. Screenshot a list of the running Docker containers that include one with the image you created

Created a docker image which is name 'linux' and show it.

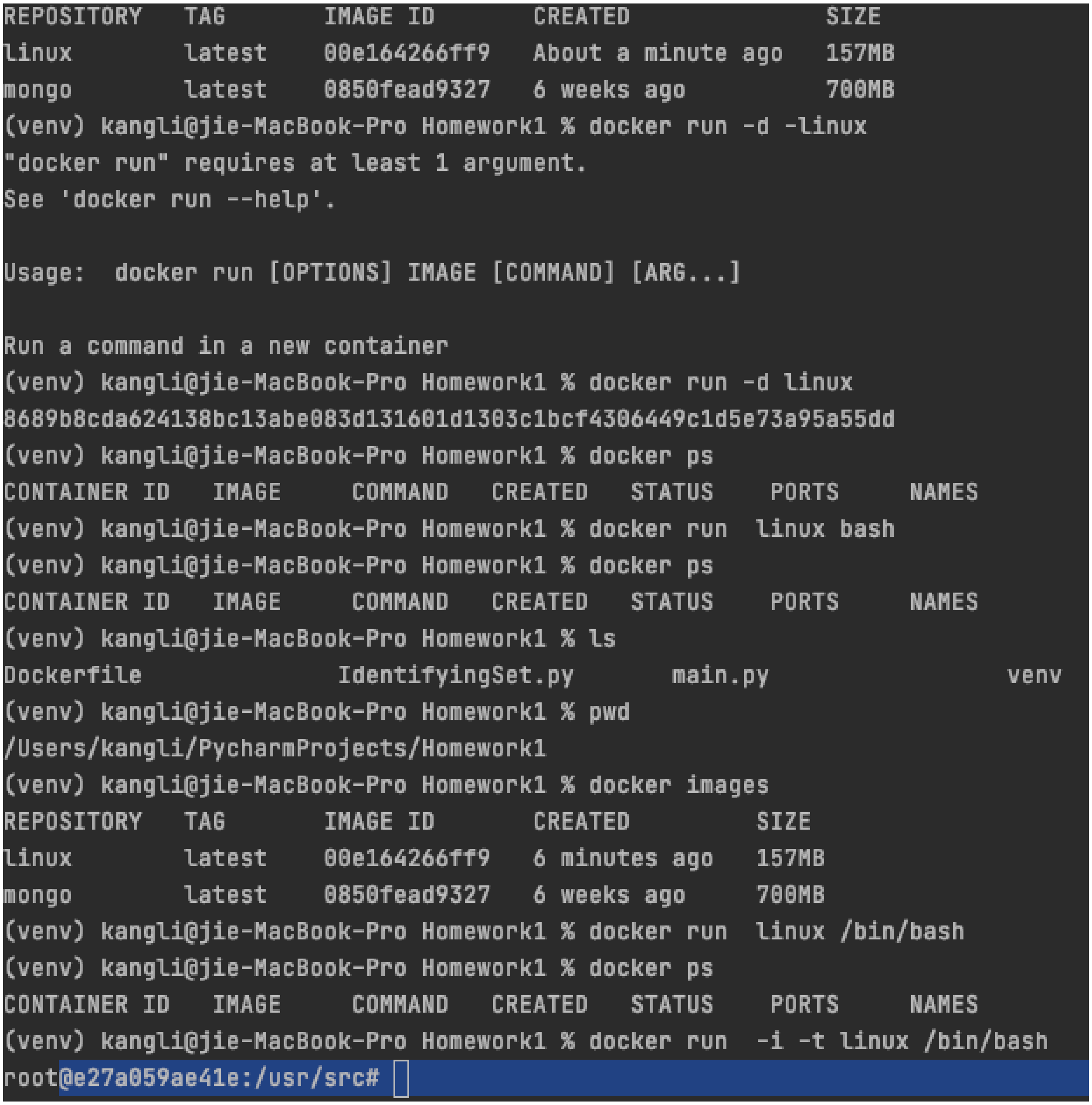


```
(venv) kangli@jie-MacBook-Pro Homework1 % docker run -d -linux
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
(venv) kangli@jie-MacBook-Pro Homework1 % docker run -d linux
8689b8cda624138bc13abe083d131601d1303c1bcf4306449c1d5e73a95a55dd
(venv) kangli@jie-MacBook-Pro Homework1 % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(venv) kangli@jie-MacBook-Pro Homework1 % docker run  linux bash
(venv) kangli@jie-MacBook-Pro Homework1 % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(venv) kangli@jie-MacBook-Pro Homework1 % ls
Dockerfile              IdentifyingSet.py       main.py                 venv
(venv) kangli@jie-MacBook-Pro Homework1 % pwd
/Users/kangli/PycharmProjects/Homework1
(venv) kangli@jie-MacBook-Pro Homework1 % docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
linux        latest    00e164266ff9   6 minutes ago   157MB
mongo        latest    0850fead9327   6 weeks ago     700MB
(venv) kangli@jie-MacBook-Pro Homework1 % docker run  linux /bin/bash
(venv) kangli@jie-MacBook-Pro Homework1 % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(venv) kangli@jie-MacBook-Pro Homework1 % docker run  -i -t linux /bin/bash
root@e27a059ae41e:/usr/src#
```

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



```
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
linux        latest    00e164266ff9   About a minute ago   157MB
mongo        latest    0850fead9327   6 weeks ago          700MB
(venv) kangli@jie-MacBook-Pro Homework1 % docker run -d -linux
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
(venv) kangli@jie-MacBook-Pro Homework1 % docker run -d linux
8689b8cda624138bc13abe083d131601d1303c1bcf4306449c1d5e73a95a55dd
(venv) kangli@jie-MacBook-Pro Homework1 % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(venv) kangli@jie-MacBook-Pro Homework1 % docker run  linux bash
(venv) kangli@jie-MacBook-Pro Homework1 % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(venv) kangli@jie-MacBook-Pro Homework1 % ls
Dockerfile              IdentifyingSet.py       main.py                 venv
(venv) kangli@jie-MacBook-Pro Homework1 % pwd
/Users/kangli/PycharmProjects/Homework1
(venv) kangli@jie-MacBook-Pro Homework1 % docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
linux        latest    00e164266ff9   6 minutes ago   157MB
mongo        latest    0850fead9327   6 weeks ago     700MB
(venv) kangli@jie-MacBook-Pro Homework1 % docker run  linux /bin/bash
(venv) kangli@jie-MacBook-Pro Homework1 % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
(venv) kangli@jie-MacBook-Pro Homework1 % docker run  -i -t linux /bin/bash
root@e27a059ae41e:/usr/src# 
```

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
- assignment 1.text 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

#why can't I see the assignment repo on github? I accepted the invitation and the repo is private, but I can't directly see the repo in my account.

//Already solved, I use git clone to pull the repo in my account as a public repo.

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.

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.

The question is to find all the unique items in the array/list. So the code file is IdentifyingSet.py, read data from the file and then traversal the file, when find the new one, add it to the set.

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.

```
Formula: counters[tuple(si)]/float(len(data))
Transform set to tule as set is not hashable.
Then calculate the occurrence with a counter.
```

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 = [S1, S2, · · · SM], which is a list of all possible sets of with unique items N

```
def all_itemsets(filename):
   f = open(filename)
   s = set()
   # dedup through set
   for line in f.readlines():
       arr = line.split(',')
       for item in arr:
           s.add(item.strip())
   ret = []
   # sort to make sure the set is uniq
   sl = sorted(list(s))

   for e in sl:
       s = set()
       s.add(e)
       t = [s]
       for item in ret:
           new = set.copy(item)
           new.add(e)
           t.append(new)

       ret += t

   return ret
```

Explanation: read the data and find the no duplicate item, then sort the set to make sure the set is unique. Then copy the item to a new one and find the new one to add it.

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 = {bread, oatmeal} occurs twice in a dataset with 100 records, then the probability of item set {bread, 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

```
def prob_S(si, data):
   # transform set to tuple as set is not hashable
   new_data = [tuple(d) for d in data]

   # calculate the occurrence with Counter
   counters = Counter(new_data)
   return counters[tuple(si)]/float(len(data))


filename = sys.argv[1]
print(cardninality_items(filename))
sets = all_itemsets(filename)
se = set()
se.add('bread')
se.add('cheese')
print(prob_S(se, sets))
```

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.

// mount the folder with the data: what is the meaning of this?

//I already downloaded the data from kaggle and the file is in the download folder. Name is : archive.zip

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.

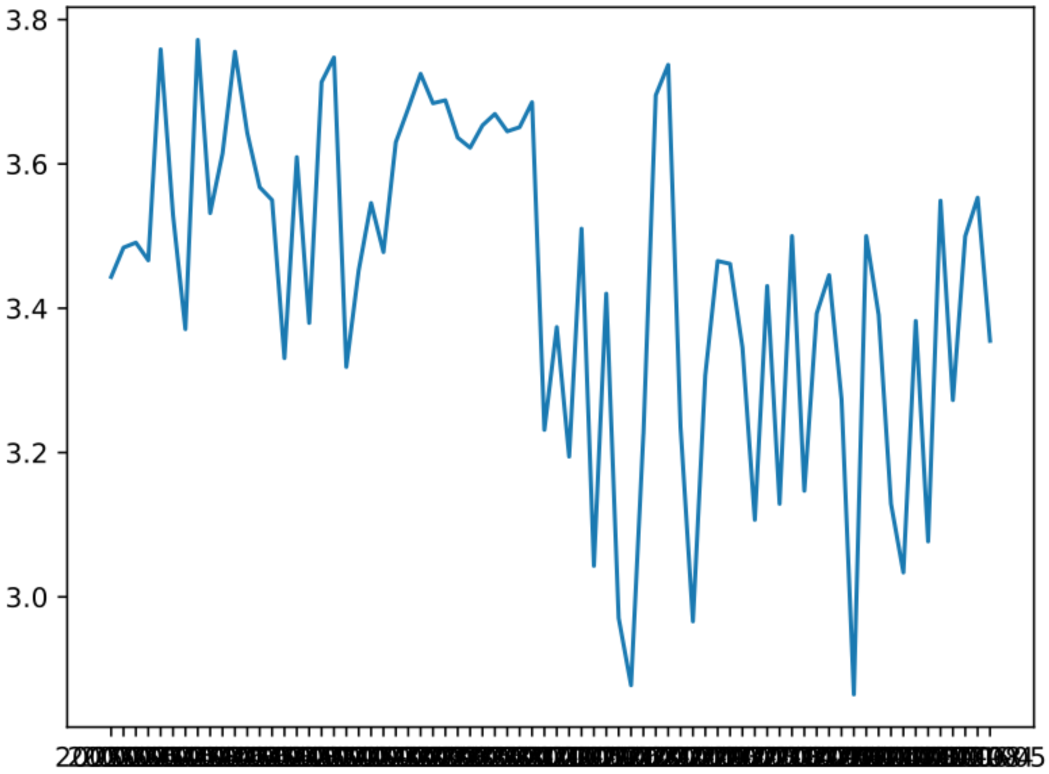
3.2 Data Analysis

Let’s answer the following questions in your writeup:

1. How many total records are there?

100,480,507

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?



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

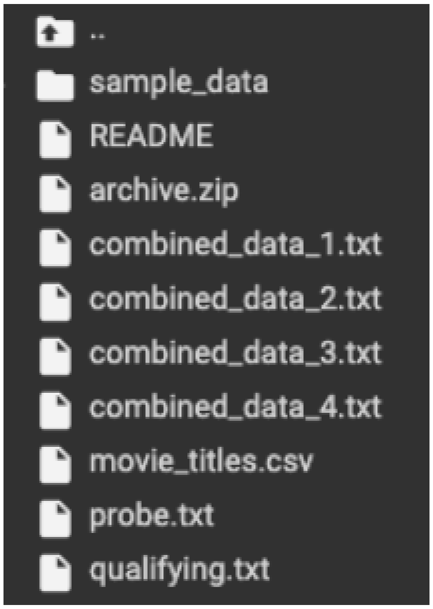
Have not yet written code to figure out this question.

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

Have not yet written code to figure out this question.

Idea: Check the rating for each film, track the rating and if the rating still active, it should be a re-released one.

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 backup your assertions with plots. Go ahead and play around with the data, and explore.



..
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

The film name changes reason.

What is majority popular films in the website per day/ month/year

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

6. What are some interesting problems that we might solve? (No need to actually solve them!)

The film name changes reason.

What is majority popular films in the website per day/ month/year

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

3. the clarity of your write-up, including

a) pipeline and code decisions,

b) perspectives on the solution,

c) and algorithmic rationale.