Data Preparation and Setup

1. Which database system(s) and version(s) are you using? How do we install it/them? (providing a link to official documentation will be enough)

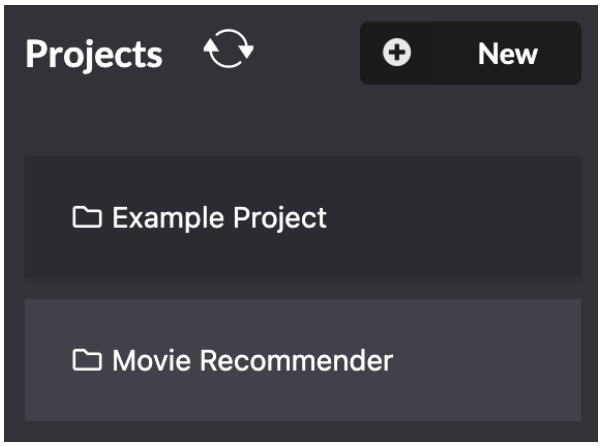
We are using Neo4j Desktop as the backend graph database. We are using Version 1.5.4 (1.5.4.99). Instructions for installation can be found here: https://neo4j.com/docs/desktop-manual/current/installation/download-installation/

2. *How do we download the data you used for your project?*

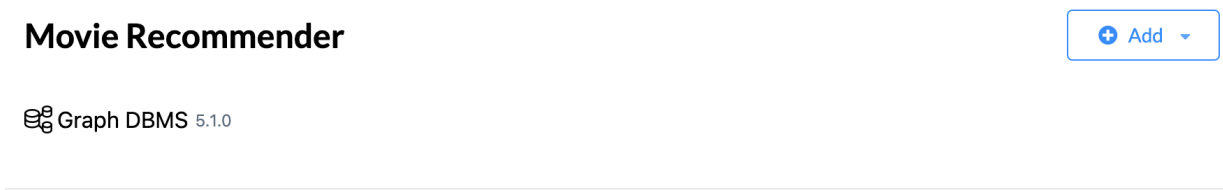
All of our cleaned and prepared data files can be found in the /data folder of our github. The data is in CSV format and takes up less than 5 MB.

3. How do we load this data into the database system?

All of the code used to ingest the data into our Neo4j database instance can be found in the /data-ingestion folder of our github. Once you open Neo4j desktop, you create a new project using the menu on the left.



Then, after activating the newly created project, click in the top right to create a new database instance.



Now, click on the settings of the new database instance and comment out this line in the config: “server.directories.import=import”

Edit settings

```
# Paths of directories in the installation.
#server.directories.data=data
#server.directories.plugins=plugins
#server.directories.logs=logs
#server.directories.lib=lib
#server.directories.run=run
#server.directories.licenses=licenses
#server.directories.metrics=metrics
#server.directories.transaction.logs.root=data/transactions
#server.directories.dumps.root=data/dumps

# This setting constrains all `LOAD CSV` import files to be under the `import` directory. Remove or
comment it out to
# allow files to be loaded from anywhere in the filesystem; this introduces possible security
problems. See the
# `LOAD CSV` section of the manual for details.
# server.directories.import=import
```

This will allow you to load data using the ingestion queries. Run the queries found in the /data-ingestion folder one by one to load all the nodes and edges into the database. You may need to change the path of the csv files in the injection queries to match the location on your machine.

5. If you are generating your own data, how do we generate it?

The scripts used to generate the data can be found in the /data-preparation folder.

Application and code

1. Which programming language(s) and version(s) are you using (Python, Java 8, C++, etc.)?

We are using Python 3.9 for our data preparation as well as our front end application.

2. List the third-party libraries needed to execute your code and how do we install them (For ex. MySQL/neo4j connector for Python)

We are using the following python libraries which can be installed via pip.
NumPy, Pandas, neo4j, and Flask.

3. If you have a GUI, how do we run it?

To run the GUI, first start up the graph database instance on Neo4j Desktop. Then, navigate to the /flaskApp folder of our github repository in your terminal and run the following command:

> flask run

The app will now be running on http://127.0.0.1:5000

Code Documentation and References

1. *Did you use some code from GitHub or other sources? If yes provide the link.*

We utilized this tutorial from Neo4j when querying our graph database from python: https://neo4j.com/developer/python/

We utilized this tutorial for creating the navbar in our Flask app: https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3

2. *If you used some online code, what changes did you make to the code?*

We used this code from the Neo4j documentation mentioned above as a skeleton for executing a query from a python file. We changed the query being used to one we wrote that generates recommendations. We also changed the connection code to connect to our own Neo4j instance.

We changed the code from the DigitalOcean tutorial to include navigation to the two distinct pages we made, the index and the recommendations page. We also changed the styling of the navbar significantly.

3. *Give a list of files in your submission which are written by you.*

Avi:

MovieRecommender/data-preparation/db-project-datacollection.ipynb
MovieRecommender/data-injection/neo4j-injestion.txt

Ritesh:

MovieRecommender/flaskApp/app.py
MovieRecommender/flaskApp/templates/recomendation.html

Ayush:

MovieRecommender/flaskApp/templates/base.html
MovieRecommender/flaskApp/templates/index.html