EARIN project Midterm report

Krzysztof Rudnicki
Jakub Kliszko

May 29, 2023

1 Progress

We have implemented reading data from csv files, preprocessing them with optional showing of some of the information about the data and used model/learner for implementing neighbour searches
Program is very flexible and allows for a lot of modification from command line arguments
Full list here:

```
options:
-h, --help                show this help message and exit
--data_limit DATA_LIMIT, -dl DATA_LIMIT
                          Specify data limit, Recommended at least 500k,
                          set to -1 for no limit
--seed SEED, -s SEED      Specify seed
--debug DEBUG, -d DEBUG
                          Use debug (more information) prints
--database DATABASE, -db DATABASE
                          Specify database path
--metric {cosine,mahalanobis,euclidean}
-m {cosine,mahalanobis,euclidean}
                          Specify metric for NearestNeighbor learner
--algorithm {auto,ball_tree,kd_tree,brute}
-a {auto,ball_tree,kd_tree,brute}
                          Specify algorithm for Nearest Neighbor learner
--anime ANIME, -an ANIME
```

```
                        Specify anime to choose
--neighbors NEIGHBORS, -n NEIGHBORS
                        Specify number of nearest neighbors
--user_threshold USER_THRESHOLD, -ut USER_THRESHOLD
                        Specify minimal number of votes
                      required for user to be included in
                      the data, set to -1 for no threshold
--anime_threshold ANIME_THRESHOLD, -at ANIME_THRESHOLD
                        Specify minimal number of votes
                        required for anime to be included
                        in the data, set to -1 for no threshold
```

2 Results

2.1 Presentation

2.1.1 Plots

2.1.2 Tables

Seed We added seed in predict function for choosing random anime, using the same seed always returns same recommendations and choosing random anime is the only random part of our code
User can specify their own seed by using -s or –seed flag by entering in command line:

```
python -s 42
```

3 Challenges

3.1 Failed attempts

Biggest challenge was realizing how overcomplicated and unnecessary difficult to implement is the first code we based on: Kaggle code with tensorflow
This solutions runs for almost 10 minutes on kaggle and implementing it to run on our local devices was a real chore that took us a good day and a half to implement
This implementation is based around very powerful Tensor Processing Unit

from google and while it is possible to change it to run on local graphics card it requires downloading both cuda and cudnn to a downgraded version supported by tensorflow (11.8) and downgrading graphics card drivers
Running it with CPU results in the model training for over 3 hours

3.2 Corrections

Suprisingly even though we based our preliminary report around different example code we managed to not make any corrections to preliminary report
All of functionality that we want to implement is available in sklearn and scipy

3.3 Results and findings

4 Finishing project

4.1 Embedding more data in user and anime

Currently we are only embedding pure rating values of users, we do not take into consideration, popularity, ”controversy”, studio which created the anime, length of anime (number of episodes and length of episodes), and when it was aired