A Movie Recommender System Using An Autoencoder in PyTorch

Participants: Jongwook Yoon and Madeline Udowa

ABSTRACT

We live in a time where everything is mass produced and shared, especially in the entertainment industry. Whether it is movies, tv shows, or series, thousands are released on a daily basis. Because of how big the entertainment industry is, it is difficult for people to keep up with what is being dropped and when, so we leave it up to streaming services to keep us up to date.

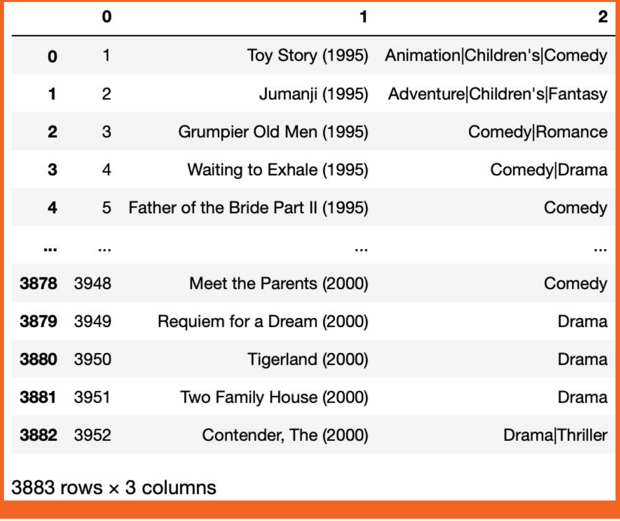
This project is a continuation of what the previous group has achieved. The previous group set out to create a recommender system model that employs collaborative filtering to suggest relevant videos to each specific user.

In this project, we will explore more ways to filter movies based on new features that we will add. What we would like to focus on is introducing the user to new movies based on their age, genre preference and ratings of other movies.

DATA SPECIFICATION

There are four different datasets that were used to aid in the production of this project.

- The first dataset is the movies dataset. In this dataset, we have the movie title, identified with ID numbers, the year they were released, and the genre of each movie. There are about 3952 movies in this dataset. See screenshot below:

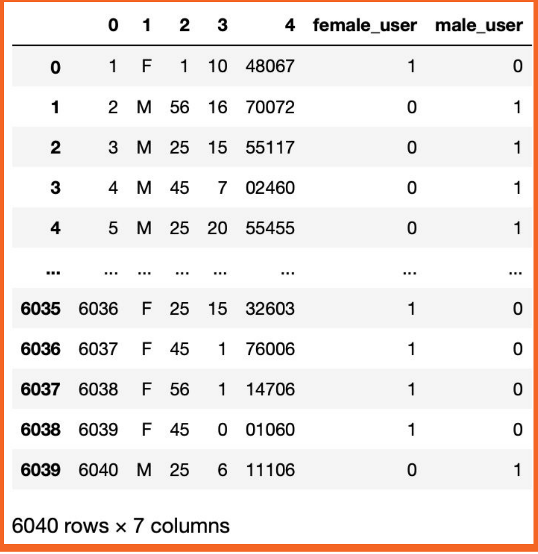


	0	1	2
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
3878	3948	Meet the Parents (2000)	Comedy
3879	3949	Requiem for a Dream (2000)	Drama
3880	3950	Tigerland (2000)	Drama
3881	3951	Two Family House (2000)	Drama
3882	3952	Contender, The (2000)	Drama|Thriller

3883 rows × 3 columns

DATA SPECIFICATION

- The second dataset is the user dataset. This dataset carries information about the user, their gender, age, occupation and zip-code. All these details about a user are accessible by the user ID. There are about 6040 users in this dataset. See screenshot below:

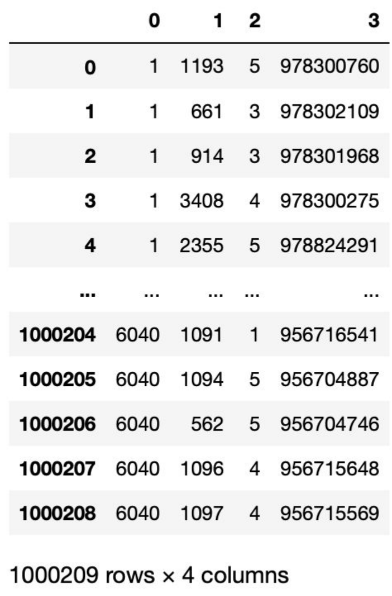


	0	1	2	3	4	female_user	male_user
0	1	F	1	10	48067	1	0
1	2	M	56	16	70072	0	1
2	3	M	25	15	55117	0	1
3	4	M	45	7	02460	0	1
4	5	M	25	20	55455	0	1
...	...	...	...	...	...	...	...
6035	6036	F	25	15	32603	1	0
6036	6037	F	45	1	76006	1	0
6037	6038	F	56	1	14706	1	0
6038	6039	F	45	0	01060	1	0
6039	6040	M	25	6	11106	0	1

6040 rows × 7 columns

DATA SPECIFICATION

- The third dataset is the ratings dataset. This dataset consists of the user’s ID, the movie ID, the rating of the movie, and timestamp that shows when the movie was rated. There are about 1000208 ratings in this dataset. See screenshot below:

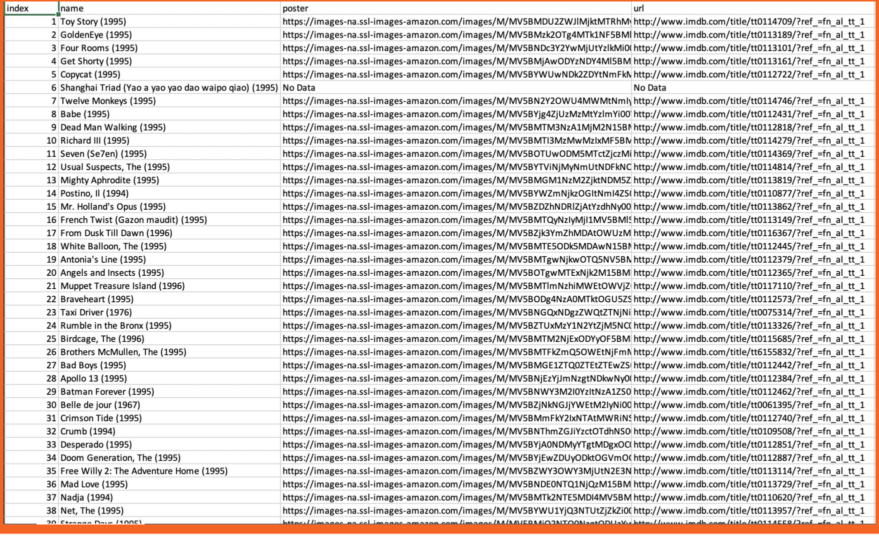


	0	1	2	3
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291
...	...	...	...	...
1000204	6040	1091	1	956716541
1000205	6040	1094	5	956704887
1000206	6040	562	5	956704746
1000207	6040	1096	4	956715648
1000208	6040	1097	4	956715569

1000209 rows × 4 columns

DATA SPECIFICATION

- The fourth dataset is another movies dataset. In this dataset, we have the movie title, identified with ID numbers, the year they were released, the poster and url links for each movie. The poster and url links will be used to visually display the movies if they have been recommended. There are about 1682 movies in this dataset. See screenshot below:



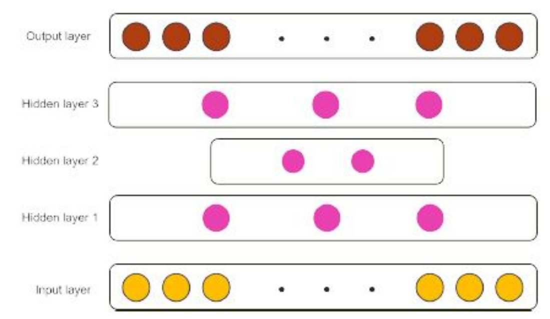
index	name	poster	url
1	Toy Story (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMDU2ZWJlMjktMTRhMy	http://www.imdb.com/title/tt0114709/?ref_=fn_al_tt_1
2	GoldenEye (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMzk2OTg4MTk1NF5BMl	http://www.imdb.com/title/tt0113189/?ref_=fn_al_tt_1
3	Four Rooms (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BNDc3Y2YwMjUtYzlkMi0	http://www.imdb.com/title/tt0113101/?ref_=fn_al_tt_1
4	Get Shorty (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMjAwODYzNDY4Ml5BM	http://www.imdb.com/title/tt0113161/?ref_=fn_al_tt_1
5	Copycat (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BYWUwNDk2ZDYtNmFkN	http://www.imdb.com/title/tt0112722/?ref_=fn_al_tt_1
6	Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)	No Data	No Data
7	Twelve Monkeys (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BN2Y2OWU4MWMtNmIy	http://www.imdb.com/title/tt0114746/?ref_=fn_al_tt_1
8	Babe (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BYjg4ZjUzMzMtYzlmYi00	http://www.imdb.com/title/tt0112431/?ref_=fn_al_tt_1
9	Dead Man Walking (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMTM3NzA1MjM2N15BN	http://www.imdb.com/title/tt0112818/?ref_=fn_al_tt_1
10	Richard III (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMTI3MzMwMzIxMF5BM	http://www.imdb.com/title/tt0114279/?ref_=fn_al_tt_1
11	Seven (Se7en) (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BOTUwODM5MTctZjczMi	http://www.imdb.com/title/tt0114369/?ref_=fn_al_tt_1
12	Usual Suspects, The (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BYTViNjMyNmUtNDFkNC	http://www.imdb.com/title/tt0114814/?ref_=fn_al_tt_1
13	Mighty Aphrodite (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMGM1NzM2ZjktNDM5Z	http://www.imdb.com/title/tt0113819/?ref_=fn_al_tt_1
14	Postino, Il (1994)	https://images-na.ssl-images-amazon.com/images/M/MV5BYWZmNjkzOGItNmI4ZS	http://www.imdb.com/title/tt0110877/?ref_=fn_al_tt_1
15	Mr. Holland's Opus (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BZDZhNDRlZjAtYzdhNy00	http://www.imdb.com/title/tt0113862/?ref_=fn_al_tt_1
16	French Twist (Gazon maudit) (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMTQyNzIyMjI1MV5BMl5	http://www.imdb.com/title/tt0113149/?ref_=fn_al_tt_1
17	From Dusk Till Dawn (1996)	https://images-na.ssl-images-amazon.com/images/M/MV5BZjk3YmZhMDAtOWUzM	http://www.imdb.com/title/tt0116367/?ref_=fn_al_tt_1
18	White Balloon, The (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMTE5ODk5MDAwN15BN	http://www.imdb.com/title/tt0112445/?ref_=fn_al_tt_1
19	Antonia's Line (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMTgwNjkwOTQ5NV5BM	http://www.imdb.com/title/tt0112379/?ref_=fn_al_tt_1
20	Angels and Insects (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BOTgwMTExNjk2M15BMl	http://www.imdb.com/title/tt0112365/?ref_=fn_al_tt_1
21	Muppet Treasure Island (1996)	https://images-na.ssl-images-amazon.com/images/M/MV5BMTlmNzhiMWEtOWVjZ	http://www.imdb.com/title/tt0117110/?ref_=fn_al_tt_1
22	Braveheart (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BODg4NzA0MTktOGU5ZS	http://www.imdb.com/title/tt0112573/?ref_=fn_al_tt_1
23	Taxi Driver (1976)	https://images-na.ssl-images-amazon.com/images/M/MV5BNGQxNDgzZWQtZTNjNi	http://www.imdb.com/title/tt0075314/?ref_=fn_al_tt_1
24	Rumble in the Bronx (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BZTUxMzY1N2YtZjM5NC	http://www.imdb.com/title/tt0113326/?ref_=fn_al_tt_1
25	Birdcage, The (1996)	https://images-na.ssl-images-amazon.com/images/M/MV5BMTM2NjExODYyOF5BMl	http://www.imdb.com/title/tt0115685/?ref_=fn_al_tt_1
26	Brothers McMullen, The (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMTFkZmQ5OWEtNjFmN	http://www.imdb.com/title/tt6155832/?ref_=fn_al_tt_1
27	Bad Boys (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMGE1ZTQ0ZTEtZTEwZS	http://www.imdb.com/title/tt0112442/?ref_=fn_al_tt_1
28	Apollo 13 (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BNjEzYjJmNzgtNDkwNy0	http://www.imdb.com/title/tt0112384/?ref_=fn_al_tt_1
29	Batman Forever (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BNWY3M2I0YzItNzA1ZS0	http://www.imdb.com/title/tt0112462/?ref_=fn_al_tt_1
30	Belle de jour (1967)	https://images-na.ssl-images-amazon.com/images/M/MV5BZjNkNGJjYWEtM2IyNi00	http://www.imdb.com/title/tt0061395/?ref_=fn_al_tt_1
31	Crimson Tide (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMmFkY2IxNTAtMWRiNS	http://www.imdb.com/title/tt0112740/?ref_=fn_al_tt_1
32	Crumb (1994)	https://images-na.ssl-images-amazon.com/images/M/MV5BNThmZGJiYzctOTdhNS0	http://www.imdb.com/title/tt0109508/?ref_=fn_al_tt_1
33	Desperado (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BYjA0NDMyYTgtMDgxOC	http://www.imdb.com/title/tt0112851/?ref_=fn_al_tt_1
34	Doom Generation, The (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BYjEwZDUyODktOGVmO	http://www.imdb.com/title/tt0112887/?ref_=fn_al_tt_1
35	Free Willy 2: The Adventure Home (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BZWY3OWY3MjUtN2E3N	http://www.imdb.com/title/tt0113114/?ref_=fn_al_tt_1
36	Mad Love (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BNDE0NTQ1NjQzM15BM	http://www.imdb.com/title/tt0113729/?ref_=fn_al_tt_1
37	Nadja (1994)	https://images-na.ssl-images-amazon.com/images/M/MV5BMTk2NTE5MDI4MV5BM	http://www.imdb.com/title/tt0110620/?ref_=fn_al_tt_1
38	Net, The (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BYWU1YjQ3NTUtZjZkZi0	http://www.imdb.com/title/tt0113957/?ref_=fn_al_tt_1
39	Strange Days (1995)	https://images-na.ssl-images-amazon.com/images/M/MV5BMjQ2NTQ0NzgtODUzYy	http://www.imdb.com/title/tt0114558/?ref_=fn_al_tt_1

DESIGN AND DEMONSTRATION

Deep Autoencoder

We made the pre-trained model using deep autoencoder

Features : 1703
(Movies: 1682 / Genre: 18 /
Gender: 2 / Age: 1)



1703

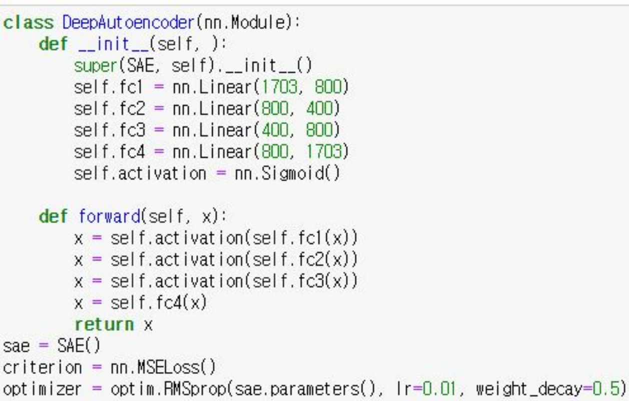
800

400

800

1703

Dimensional reduction



```
class DeepAutoencoder(nn.Module):
    def __init__(self, ):
        super(SAE, self).__init__()
        self.fc1 = nn.Linear(1703, 800)
        self.fc2 = nn.Linear(800, 400)
        self.fc3 = nn.Linear(400, 800)
        self.fc4 = nn.Linear(800, 1703)
        self.activation = nn.Sigmoid()

    def forward(self, x):
        x = self.activation(self.fc1(x))
        x = self.activation(self.fc2(x))
        x = self.activation(self.fc3(x))
        x = self.fc4(x)
        return x
sae = SAE()
criterion = nn.MSELoss()
optimizer = optim.RMSprop(sae.parameters(), lr=0.01, weight_decay=0.5)
```

Tool: Jupyter notebook / Framework: Pytorch

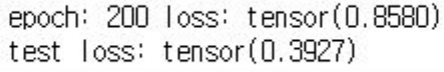
DESIGN AND DEMONSTRATION

Deep Autoencoder

Results of training and testing

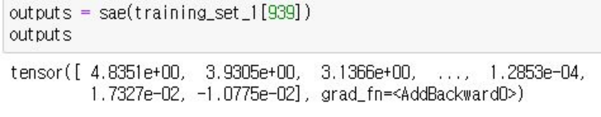
We have to find and choose hyper-parameters(epoch, layers, etc) to get a model has low loss

Epoch: 200 loss: RMSE(Root mean square error)



```
epoch: 200 loss: tensor(0.8580)
test loss: tensor(0.3927)
```

Sae is the pre-trained model, and it shows a predicted result



```
outputs = sae(training_set_1[939])
outputs
```

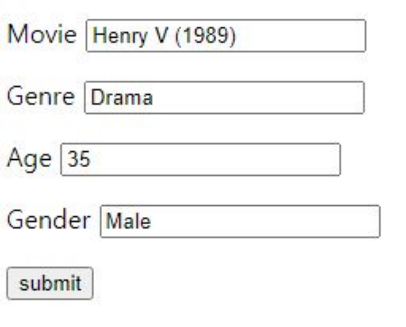
```
tensor([ 4.8351e+00,  3.9305e+00,  3.1366e+00,  ...,  1.2853e-04,
         1.7327e-02, -1.0775e-02], grad_fn=<AddBackward0>)
```

DESIGN AND DEMONSTRATION

Movie Recommender

We can recommend top N movies

Tool: pycharm Framework: Flask

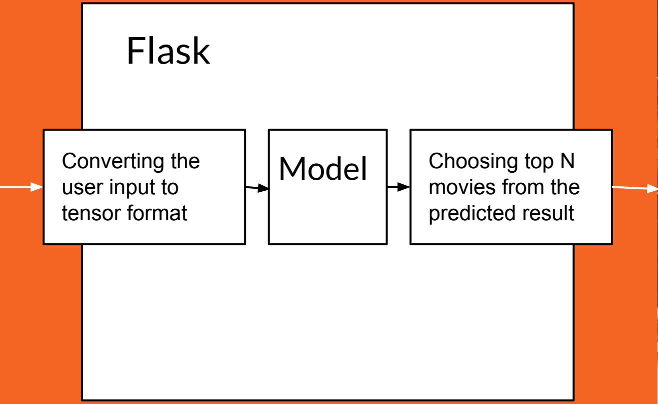


Movie: Henry V (1989)

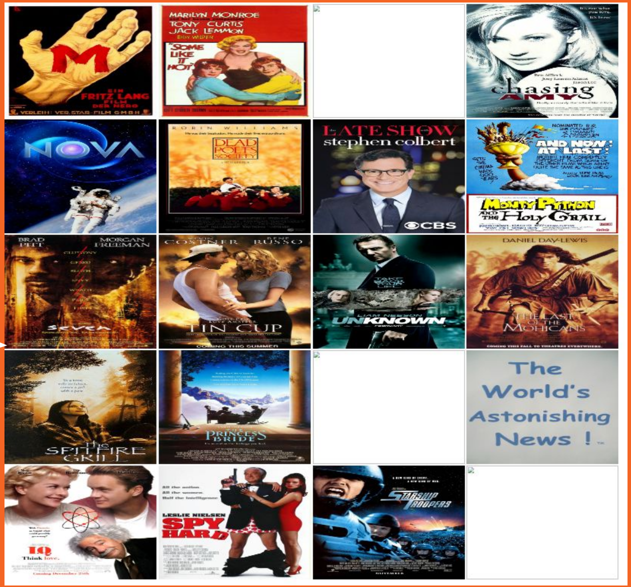
Genre: Drama

Age: 35

Gender: Male



Flask
Converting the user input to tensor format
Model
Choosing top N movies from the predicted result

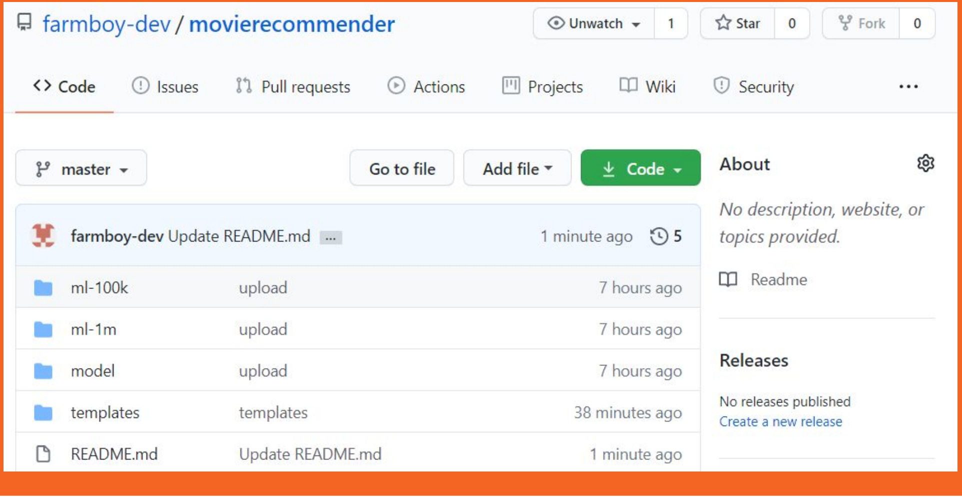


*Blank means there is no poster data

REPOSITORY / ARCHIVE

Github

https://github.com/farmboy-dev/movierecommender



farmboy-dev Update README.md … 1 minute ago 5

ml-100k	upload	7 hours ago
ml-1m	upload	7 hours ago
model	upload	7 hours ago
templates	templates	38 minutes ago
README.md	Update README.md	1 minute ago

No description, website, or topics provided.

No releases published
Create a new release

What’s next?

1. Training the model has better performance.
2. Using latest dataset such as IMDB and TMDB
3. (optional) we can use movie reviews

Thank you