Movielens 100k – Recommendtion systems

This document consolidates all the results that we received after running the jupyter notebook, which contains the code for this project.

In the course of the project, we compared the different models and combinations of different parameters. The comparison of the quality of the model is measured according to the MAE we received from the test group.
Therefore, if a document is registered with an MAE without reference to a training group or a test group - the reference is to the test group.

Exercise 1:

A. Code for MF - matrix factorization and GMF - General Matrix Factorization is attached to the code in jupyter notebook.

B. We were asked to calculate the MAE and the runtime of the test group when we change the dimension of the embedding layer and the size of the drop-out

For MF - we performed the tests with the following options (total 4):

Embedding:

- Item – user
- User – item

Drop-out:

- 0.1
- 0.5

Below is a summary of the results we received for the MF for the model training and for the TRA and MAE of the test group:

Embedding	Dropout	Mean Absolute Error - train	Time	Mean Absolute Error - test
Item - User	0.5	1.8017	200.0921	1.8032
Item - User	0.1	0.8012	233.1664	0.8021
User - item	0.5	1.7993	206.7634	1.8085
User -item	0.1	0.8022	231.2722	0.8035

Best result for MF:
• In terms of MAE of the test group:
Combination of embedding = Item - User, drop-out = 0.1
• In terms of times of model training: combination of embedding = item-user, drop-out = 0.5
• We will treat the result with the MAE of the lowest test group as better because what interests us is that the prediction will be as good as possible.

For GMF - General Matrix Factorization -we performed the tests with the following options (total 2):

Embedding:

- User – item

Drop-out:

- 0.1
- 0.5

Below is a summary of the results we received for the GMF for the model training and for the MAE and TRAIN of the test group:

Embedding	Dropout	Mean Absolute Error - train	Time	Mean Absolute Error - test
User - item	0.5	0.90155	338.712	0.748837
User - item	0.1	0.9359	72.9278	0.7613

Best result for GMF:
• In terms of MAE of the test group:
Combination of embedding = user-item, drop-out = 0.5
• In terms of times of model training:
Combination of embedding user-item, drop-out = 0.1
We will treat the result with the lower MAE as better because what interests us is that the prediction will be as good as possible.

B. Now we will compare the different results -

Let's take the good results of MF and GMF and compare to the non-personal recommendation we wrote in Job 2 based on the average rating of the films.

It should be noted that we treat the best result by measurement with the MAE as little as possible and not the lowest time measurement because as the MAE is smaller, it indicates that the distance from the test results to that of the smallest workout - the best prediction.

Another observation that we can see is that the model (combination of parameters) that gave MAE with the best result during model training was also the one that gave the best result of the MAE to the test group, .but the duration of training and speed is not related

The following table summarizes the best results of the various models:

Model	MAE - train	Time	Mae - test
Matrix Factorization	0.8012	233.1664	0.8021
Generalized Matrix factorization User-item 0.5	0.90155	338.712	0.748837
Non-Personalized (HW2)	-	486.9826	1.0196

We will scale the models according to the high precision (low MAE) to low (MAE high) according to the MAE of the test group:

1. GMF
2. MF
3. non-Personalized

We will scale the models according to the high precision (low MAE) to low (MAE high) according to the MAE of the training group:

1. MF
2. .MFG
3. non-Personalized - -missing

We will scale the models by the duration of the model's training period (from short to long):

1. MF
2. GMF
3. non-Personalize

Exercise 2:

A. Code for CFG in the jupeter notebook

B. We calculated the MAE values and the training time - for a combination of different parameters:

Size of Layer: 8, 16, 50, 100

Optimizer = sgd, adam, Nadam, Adadelta

Function Loss = mean absolute error, mean squared error, mean_absolute_percentage_error

Activation Function = relu, tanh, linear, selu

drop-out of: 0.1, 0.5

Below is a summary of the different results for each of the parameters:

- **Size of Layer**

  We will notice that as we increase the size of the project layer, the model's accuracy increases, so we will run the rest with 100

Size Of layer	Time	MAE - test
8	46.7975	1.8366
16	42.6713	1.8234
50	82.5716	1.8218
100	70.3447	1.8201

- **Optimizer**

We will notice that when the optimization function is of **adam** the accuracy of the model is increased

Optimizer	Time	MAE - test
sgd	133.4138	1.8201
adam	53.6670	1.818
Nadam	97.4659	1.8374
Adadelta	79.3800	1.8270

- **Function loss**

We will note that for the calculation of the loss function: **MAE**, the accuracy level is the highest, so the other tests will use **MAE**

Function loss	Time	MAE - test
mean absolute error	47.21113	1.7840
mean squared error	156.1635	1.8211
mean_absolute_percentage_error	41.9661	1.9701

- **Activation Function**

We observe that when we use **linear** accuracy is the highest

Activation Function	MAE-Train	Time	MAE - test
relu	1.7417	64.8764	1.8126
tanh	2.5359	104.3312	2.5383
linear	1.7611	41.9993	1.7660
selu	1.7610	44.3085	1.7997

- **Drop-out**

We will notice that for the **drop-out = 0.1** the accuracy is the highest and therefore the rest of the tests will use **0.1**

Drop-out	MAE - train	Time	MAE - test
0.1	0.7470	77.6727	0.7967
0.5	1.7712	37.4019	1.8371

The parameters that yielded the best results (each in its field) in each of the parameters are for Exercise 2 in terms of the best MAE of the test group**:**

Size of Layer100

Optimizer = adam

Function Loss = mean absolute error

Activation Function = linear

drop-out of: 0.1

We will run again with the combination of all the parameters that brought the best in their field the best result and see if it improves the results of one of them:

MAE - train	Time	MAE - test
0.9445	32.6114	0.9512

We note that this combination resulted in a less favorable result than one of the measurements.

Therefore, the best result came from the combination of parameters:

Size of Layer100

Optimizer = nadam

Function Loss = mean absolute error

Activation Function = relu

drop-out of: 0.1

And therefore we recommend using their combination that actually yielded the best result since we performed a refinement phase here and found the parameters that will yield the best result for the information on which we are modeling the model.

C. In conclusion:

Now we'll compare all the models we've used so far and see who has the best results:

Model	MAE - train	Time	MAE - test
Matrix Factorization	0.8012	233.1664	0.8021
Generalized Matrix factorization	0.90155	338.712	0.748837
Non-Personalized (HW2)	לא חושב - בעבודה 2	486.9826	1.0196
NCF	0.7470	77.6727	0.7967

We will scale the models according to the high precision (low MAE) to low (MAE high) according to the MAE of the test group:
1. GMF
2. NCF
3. MF
4. Non-Personalized

We will scale the models according to the high precision (low MAE) to low (MAE high) according to the MAE of the training group:
1. NCF
2. MF
3. GMF
4. Non-Personalized

We will scale the models by the duration of the model's training period (from short to long):
1. . NCF
2. MF
3. GMF
4. Non-Personalized

We received that GMF model produces more accurate results than any other model.
We will note that NCF also has good results immediately after.
It is also important to remember that of the two models we spent a lot of time in refinement and finding the correct NCF parameters until we reached the above accuracy, so it is possible that with another work the GMF model may also get more accurate results.

Exercise 3:

The model that will be built now will be a context-based model - we will recommend a movie to the user according to the user's characteristics (age and gender)

For this exercise we will add and use the information that exists on the users.
I chose to use age and gender because these are things that I think are more influential than the human profession.
Age of the user can be a significant factor for young ages, for example will recommend more children's movies and adventures than horror movies, and the user's sex can indicate a genre favorite.

A preliminary step to build the model with the new features will be data processing done by one of the data into a single dataframe that will contain the user ID, identifier of the film that he rated, rating the film, user gender, age of the user so that we can link the information in the tables Variance.
Another preliminary step that we had to make in order for the model to accept the user's gender was to convert it to a numerical value so that we could insert it into the model that was built.

We chose to implement the model by NCF because it is the model we found from the four models that got the good results right after the GMF but we chose to use it.

The model we are implementing is a hybrid model of feature combination as we learned in this class because we use additional features on the user and combine them together with information about the rating of the film.

The model's architecture is based on the idea of feature augmentation - the characteristic that the second recommendation system (based on the new features) uses is the output of the first-user system.

We want to build an NCF model that will take several layers, first it will combine the embedding layers that are the user and then the item by using Concatenate, we will output the user-item output to the input of the next layer, the viewer's gender or the viewer's age, and this will be our hybrid model.

Also remember before training to split the information to 80 percent for the training group and 20 percent for the test group.

B. Code in the jupyter attached to the submission.

C. We added gender and age layers and trained them with values of the same parameters that we defined as default - the values that produced the best results for the NCF model from Exercise 2:

- **Size of Layer**100
- **Optimizer** = Nadam
- **Function Loss** = mean absolute error
- **Activation Function** = relu
- **drop-out of**: 0.1,

In addition, we calculated with additional parameters and combinations to see if here too we can achieve optimal integration for the new model:

Size of Layer:, 20, 100

Optimizer = adam, Nadam

Function Loss = mean absolute error, mean squared error

Activation Function = relu, linear, selu

drop-out of: 0.1, 0.5

The following table summarizes all measurement results for the new models:

Additional Feature	parameter	MAE- train	TIME	MSE - test
Age	default	0.91272	104.7925	0.9669
Gender	default	0.9243	106.7996	1.0132
Age	Adamx	0.9138	99.2087	0.9384
Gender	adamx	0.9168	155.6402	0.9246
Age	linear	0.91791	112.5583	0.9872
Gender	linear	-	113.3625	0.9757
Age	Selu ,50, mse	0.9104	150.9089	0.9298
Gender	Selu,50, mse	-	104.8177	1.0013
Age	Mse	-	106.4828	0.9899
Gender	mse	-	107.7074	0.9613
Age	0.5	0.9039	134.1343	0.9247
Gender	0.5	0.92639	138.3305	0.9618

The best results for the above models:
:

Additional Feature	parameter	MAE- train	TIME	MSE - test
Age	0.5	0.9039	134.1343	0.9247
Gender	adamx	0.9168	155.6402	0.9246

D. Now we will compare all the different models we have tested over the above project and see who has produced the best results:

Model	MAE - train	Time	MAE - test
Matrix Factorization	0.8012	233.1664	0.8021
Generalized Matrix factorization	0.90155	338.712	0.748837
Non-Personalized (HW2)	לא חושב - בעבודה 2	486.9826	1.0196
NCF **תרגיל 2**	0.7470	77.6727	0.7967
NCF + age	0.9039	134.1343	0.9247
NCF + gender	0.9168	155.6402	0.9246

We will scale the models according to the high precision (low MAE) to low (MAE high) according to the MAE of the test group:

1. GMF
2. NCF
3. MF
4. NCF + gender
5. NCF + age
6. Non-Personalized

We will scale the models according to the high precision (low MAE) to low (MAE high) according to the MAE of the training group:

1. NCF
2. MF
3. GMF
4. NCF + gender
5. NCF + age
6. Non-Personalized

We will scale the models by the duration of the model's training period (from short to long):
1. NCF
2. NCF + age
3. NCF + gender
4. MF
5. GMF
6. Non-Personalized

E. After going over all the different results and models - we can recommend the GMF model because it received the lowest MAE score for the test group, we note that the NCF of Exercise 2 received the best MAE during model training and the best running time.

Since in the recommendation system it is important to us both the combination of calculation time and accuracy we understand that GMF received the smallest MAE but the longest of the new models, however, the accuracy is better but it comes with the tradeoff of waiting for the recommendation.

Given the combination of time and accuracy of the test group in my opinion to recommend the NCF of Exercise 2 as the recommended system.

Throughout the various models we have seen how using incorrect parameters can have a significant impact on training results. We have also seen cases in which one parameter doubled the MAE score in both the test group and the training group.

Therefore, in my opinion, because in stages 2 and 3 we performed the most tests and combinations of different parameters, we came to the best parameter combination since we can see in the results of Exercise 2 that there are measurements in which the MAE is significantly larger than the main models in which we used the combination of the parameters that produced the best result.

In addition, the more successful the NCF hybrid model will be, the more successful the success will be, as the 80,000 ratings are not enough to form a good opinion of what movie a woman will love, or what film to recommend to Ben 80.