Course Project

Form:	Jupyter notebook file including images and text explanation
Language:	English
Requirements:	The report should be clear, readable and include all code documented
Submission:	.ipynb file via Moodle. The file name should include the students' ids
Contact:	
Deadline for submission:	**February 21, 2021**

Each student will submit his\her own assignment.

Submit your solution in the form of an Jupyter notebook file (with extension ipynb). Images of graphs or tables should be submitted as part of the notebook itself. The code used to answer the questions should be included, runnable and documented in the notebook. Python 3.6 or higher should be used.

The goal of this project is to let you practice in a data scientist daily work by leveraging recommender systems algorithms you learnt in the course and customize them in order to solve real business problems.

Submission: Submission of the project will be done via Moodle by uploading a Jupyter notebook file containing code, plots and explanations. The homework needs to be entirely in English. The deadline for submission of Homework 1 is set to February 14, 2021 end of day Israel.

We will use a dataset based on the MovieLens 1M rating dataset after some pre-processing to adapt it to an implicit feedback use case scenario. You can download the dataset used by this implementation of the paper Neural Collaborative Filtering or from the NeuralCollaborativeFiltering_implicit notebook in Moodle.

Question 1: Matrix Factorization with custom loss (35 points)

You work for an e-commerce company "Buy Here". You are using the Matrix Factorization algorithm to recommend consumers with products which may be relevant to them using implicit feedback. The product manager tells you that he wants to improve the accuracy of the prediction model for more expensive products since these products are more profitable to the company.

One of the common techniques to be more accurate for more expensive products is to give a higher weight in the loss function to more expensive products. You are using the Binary Cross Entropy loss function which is suitable for binary classification problems. Here is the custom loss function, when adding a weight to each instance in the training data, to give different weight for each sample.

$$L = -\frac{1}{N}\sum_{(i,j)\in S}^{N} \alpha_j \left(y_{i,j} \log\left(\sigma\left(\mu + p_i + o_j + \boldsymbol{u^T}_i \boldsymbol{v_j}\right)\right) + (1 - y_{i,j}) \log(1 - \sigma\left(\mu + p_i + o_j + \boldsymbol{u^T}_i \boldsymbol{v_j}\right)\right))$$

Here,

α_j is the weight for instances which involve item *j*, $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic function, μ is the global bias, p_i, o_j are the user and item bias respectively and $\boldsymbol{u_i}, \boldsymbol{v_j}$ are column vectors with size *K*, representing the latent weight vector of the user *i* and item *j* respectively. *N* is the number of instances in the training dataset.

a. Derive the Gradient Decent update rule for the user and item latent vector weights as well as for the global bias, user bias and item bias variables. Explain each step. (30 points)
b. Given the prices of the different items from the catalog. How will you set the weights for each training instance based on the item price. You can offer more than one alternative. Explain your suggestion, and the advantage of each choice. (5 points)

Question 2: Neural Collaborative filtering implementation (50 points)

For this question, you will use the item_price.csv file and the original dataset, implement and evaluate a price sensitive recommender system

a. Use the item_price.csv file to get the prices of each item. Explore the price distribution of items. (5 points)
b. To evaluate the performance of the price sensitive model we will add another metric Revenue@*K* which will measure the overall revenue from the top 5 recommended hits. Implement the metric. (5 points)
   The function will be calculated as follows:
   For each user: sum the prices of the top *K* recommended items which were rated as the revenue from the user.
   Calculate the mean revenue from all users.
c. Suggest a metric of your own which will incorporate both the ranking of the recommended items as well as its price. Explain why this metric is suitable and demonstrate it as part of the evaluation in point e below. (10 points)
d. Select one of the models presented in the Neural Collaborative Filtering paper and incorporate the movie price to the loss function as part of training. (10 points)
e. Compare the results of the original model and the one with the customized loss across the four metrics: MRR@5, NDCG@5,Revenue@5 and your custom metric. Compare between different heuristics of item price to weights mapping. Present the comparison results, discuss the results and the trade-offs and optimize. Verify and present that the learning is 'healthy' (no overfitting, no under-fitting and that the results make sense). (20 points)

Question 3: Hybrid recommender systems (15 points)

Cold start or users\items with a small number of interactions is a very common scenario in real world. In this question you will plan how you can leverage content based features to handle the cold start scenario.

a. Take a look at the original MovieLens 1M dataset. Which user and movie features could you use to enhance your recommender system and provide effective recommendations to users or items with a small number of ratings. (5 points)
b. Describe a neural network based model to incorporate user or movie related features to the recommender system. Explain your suggestion. (there is no need to implement, provide pseudo code\visual). (5 points)
c. How will you incorporate movie genres into the recommender system? How will you handle movies which belong to multiple genres? Explain the challenge and the proposed solution (there is no need to implement, provide pseudo-code). (5 points)

Good luck