Questionnaire (8th chapter)

Total de pontos 0/25

O e-mail do participante (**breno.xavier@ufpe.br**) foi registrado durante o envio deste formulário.

✗ What problem does collaborative filtering solve? * .../1

Collaborative filtering solves the problem of recommending items or products to users based on their preferences and behaviors, by analyzing their past interactions and finding patterns or similarities with other users. It is commonly used in recommendation systems for movies, books, music, and other products or services

✗ How does it solve it? * .../1

Collaborative filtering solves the problem of recommending items to users based on their past interactions and the behavior of similar users

✗ Why might a collaborative filtering predictive model fail to be a very useful recommendation system? * .../1

A collaborative filtering predictive model might fail to be a very useful recommendation system when there are insufficient or poor quality ratings and when there is a cold start problem

✗ What does a crosstab representation of collaborative filtering data look like? * .../1

A crosstab representation of collaborative filtering data is a matrix where rows correspond to users, columns correspond to items, and each cell contains a rating or a measure of user-item interaction such as the number of clicks, purchases, or views. The matrix is usually sparse, meaning that most cells are empty or have missing values

✗ What is a latent factor? Why is it "latent"? * .../1

A latent factor is a variable that is not directly observed but is inferred from other variables and data

✗ What is a dot product? Calculate a dot product manually using pure Python with lists. * .../1

dot product is a mathematical operation between two vectors that results in a scalar value. To calculate the dot product between two vectors a and b, you multiply each corresponding element of the vectors together, then sum the products

```
vector1 = [8, 9, 15]
vector2 = [88, 99, 101]

dot_product = sum([vector1[i] * vector2[i] for i in range(len(vector1))])
print(dot_product)  # saida: 3110
```

✗ What does `pandas.DataFrame.merge` do? * .../1

it is a method used to merge two or more dataframes into a single dataframe. It combines the rows of the two dataframes based on a common column or index, and can perform inner, outer, left, or right joins depending on the specified merge method. It is a flexible and powerful tool for data manipulation and can handle merging of dataframes with different shapes and structures

✗ What is an embedding matrix? * .../1

An embedding matrix is a matrix that maps categorical variables to a vector space, enabling their use in machine learning algorithms

✗ What is the relationship between an embedding and a matrix of one-hot-encoded vectors? * .../1

An embedding is a dense representation of a categorical variable, while a matrix of one-hot-encoded vectors is a sparse representation. The embedding matrix is a learned matrix used to map sparse vectors to dense embedding

✗ Why do we need `Embedding` if we could use one-hot-encoded vectors for the same thing? * .../1

We need Embedding because one-hot-encoded vectors are too large and sparse for large vocabularies

✗ What does an embedding contain before we start training (assuming we're not using a pretained model)? * .../1

An embedding contains random values before training (assuming we're not using a pretrained model), which get updated during training through backpropagation to capture meaningful representations of the input

✗ What does `x[:,0]` return? * .../1

Returns all the elements in the first column of the two-dimensional NumPy array

✗ What is a good loss function to use for MovieLens? Why? * .../1

For MovieLens dataset, a good loss function to use is mean squared error (MSE) because it heavily penalizes large errors, which is important for recommendation systems where the goal is to accurately predict ratings. Additionally, MSE is a common loss function for regression problems and is easy to implement

✗ What would happen if we used cross-entropy loss with MovieLens? How would we need to change the model? * .../1

Using cross-entropy loss with MovieLens would not make sense since the ratings are not categorical; to use it, we would need to binarize the ratings and modify the output layer

✗ What is the use of bias in a dot product model? * .../1

The bias term in a dot product model allows the model to make predictions even if all input features are zero

✗ What is another name for weight decay? * .../1

L2 regularization

✗ Why does reducing weights lead to better generalization? * .../1

Reducing weights reduces model complexity, which in turn reduces overfitting and improves generalization

✗ What does `argsort` do in PyTorch? * .../1

Returns the indices that would sort the input tensor

✗ Does sorting the movie biases give the same result as averaging overall movie ratings by movie? Why/why not? * .../1

No, because movie biases and overall movie ratings measure different things: the former measures how much people tend to like or dislike a given movie relative to others, while the latter measures the average rating of the movie across all users

✗ How do you print the names and details of the layers in a model? * .../1

You can print the names and details of the layers in a model by calling the summary() method on the model object. Alternatively, you can print the layer names and parameters by iterating over the layers in the model using a for loop and printing the relevant information for each layer

✗ What is the "bootstrapping problem" in collaborative filtering? * .../1

The "bootstrapping problem" in collaborative filtering refers to the difficulty of making recommendations for new users or items with little to no existing data

✗ How can feedback loops impact collaborative filtering systems? * .../1

Feedback loops can reinforce existing biases in collaborative filtering systems by recommending similar items to users based on their past behavior, which can lead to a lack of diversity in recommendations

✗ When using a neural network in collaborative filtering, why can we have different numbers of factors for movies and users? * .../1

In collaborative filtering with neural networks, we can have different numbers of factors for movies and users because they can have different degrees of complexity and information

✗ Why is there an `nn.Sequential` in the `CollabNN model`? * .../1

It is used to stack multiple layers and modules sequentially in the CollabNN model

✗ What kind of model should we use if we want to add metadata about users and items, or information such as date and time, to a collaborative filtering model? * .../1

To add metadata about users and items, or information such as date and time, to a collaborative filtering model, we should use a hybrid collaborative filtering model, which combines collaborative filtering with other types of models such as content-based filtering or matrix factorization with side information