Recommender System Machine Learning Project for Beginners-4

Project Overview

Business Objective

We often ask our friends with the same tastes to recommend products to use, movies to watch, and whatnot. We trust the recommendations of friends whose taste is similar to ours. Collaborative filtering does the same job. Collaborative filtering is primarily focused on finding similarities between users and mutually encouraging user preferences. Collaborative filtering is the predictive process behind the recommendation engine. The recommendation engine analyzes information about users with similar tastes and assesses the target person's likelihood of enjoying something. Collaborative filtering uses various algorithms to filter data from user reviews and provide personalized recommendations to users with similar settings. Collaborative filtering uses interactions and data collected from other users to filter information.

As a part of a series of Recommender system projects, this project covers Recommendations using a wide variety of Collaborative Filtering algorithms in Python. If you haven't already visited, here is the previous project of the series Recommender System Machine Learning Project for Beginners-3

Data Description

This is a transactional data set containing the transactions for a UK-based non-store online retail. The company mainly sells unique all-occasion gifts.

Aim

To build a Recommender system using various Model-Based and Memory-based collaborative filtering techniques

Tech Stack

- ➔ Language: Python
- ➔ Libraries: sklearn, surprise, pandas, matplotlib, scipy, numpy, pickle

Approach:

1. Data Description
2. Data cleaning
3. Memory-based approach
    a. User-to-User collaborative recommendation
    b. Item-to-Item collaborative recommendation
4. Models:
    a. KNN model
    b. Non-negative Matrix Factorization (NMF)
    c. Co-Clustering model
5. Evaluation metrics:
    a. Root mean square error
    b. Mean absolute error
    c. Cross-validation score

Modular code overview:

```
input
|__ Rec_sys_data.xlsx

lib
|__ Collaborative_Filtering.ipnyb

src
|__ engine.py
|__ ml_pipeline
    |__ model.py
    |__ preprocessing.py
    |__ utils.py

output
|__ CoClustering_model.pkl
|__ nmf_model.pkl

requirements.txt
```

Once you unzip the modular_code.zip file, you can find the following folders within it.

1. input

2. src
3. output
4. lib
5. requirements.txt

1. The input folder contains the data that we have for analysis. In our case, it contains Rec_sys_data.xlsx

2. The src folder is the heart of the project. This folder contains all the modularized code for all the above steps in a modularized manner. It further includes the following.
    a. ML_pipeline
    b. engine.py

The ML_pipeline is a folder that contains all the functions put into different python files, which are appropriately named. These python functions are then called inside the Engine.py file.

3. The output folder contains the best-fitted model that we trained for this data. This model can be easily loaded and used for future use, and the user need not have to train all the models from the beginning.

4. The lib folder is a reference folder, and it contains the original ipython notebook that we saw in the videos.

5. The requirements.txt file has all the required libraries with respective versions. Kindly install the file by using the command pip **install** -r **requirements**.**txt**

Project takeaways

1. What are Collaborative filtering and its type?
2. What is a Memory-based approach?
3. What are User-to-User Collaborative Filtering and its implementation?
4. What are Item-to-Item Collaborative Filtering and its implementation?
5. What is cosine similarity?
6. What is a Model-based approach?
7. How to perform a model-based approach using KNN?
8. What is Matrix Factorization?
9. How to perform a model-based approach using Matrix Factorization?

10. Understanding Surprise library in detail
11. What are different prediction algorithms present in the surprise library?
12. What is the surprise library's Non-negative matrix factorization(NMF) model?
13. How to implement Non-negative matrix factorization(NMF)?
14. What is the Co-Clustering model in the surprise library?
15. How to implement a Co-Clustering model?
16. How to evaluate a recommendation system using the surprise library?
17. What is a CSR matrix?
18. Understanding Cross-validation