TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
NGÀNH CÔNG NGHỆ THÔNG TIN

BÁO CÁO TIẾN ĐỘ -TUẦN 04
MÔN: ĐỒ ÁN 3
ĐỀ TÀI: TÌM HIỂU VỀ THUẬT TOÁN RECOMMENDATION

GVHD : Thầy Huỳnh Xuân Phụng
SVTH :
Nguyễn Thành Như **17110202**
Võ Ngọc Thuận **17110234**

TP. Hồ Chí Minh, tháng 10 năm 2020

1. Demo Recommendation System with Collaborative Filtering

1.1. Code Demo

```python
import pandas as pd
from scipy import sparse
from sklearn.metrics.pairwise import cosine_similarity
ratings=pd.read_csv("toy_dataset.csv",index_col=0)
ratings.fillna(0, inplace=True)
ratings
print(ratings)
print("==================1==============")
def standardize(row):
    new_row = (row - row.mean())/(row.max()-row.min())
    return new_row
df_std = ratings.apply(standardize).T
print(df_std)
sparse_df = sparse.csr_matrix(df_std.values)
corrMatrix = pd.DataFrame(cosine_similarity(sparse_df),index=ratings.columns,columns=ratings.columns)
print(corrMatrix)
print("==================2==============")
corrMatrix = ratings.corr(method='pearson')
print(corrMatrix.head(6))
print("==================3==============")
def get_similar(movie_name,rating):
    similar_score = corrMatrix[movie_name]*(rating-2.5)
    similar_score = similar_score.sort_values(ascending=False)
    #print(type(similar_ratings))
    return similar_score
action_lover = [("action1",5),("romantic2",1),("romantic3",1)]
similar_scores = pd.DataFrame()
for movie,rating in action_lover:
    similar_scores = similar_scores.append(get_similar(movie,rating),ignore_index = True)
print(similar_scores.head(10))
print("==================4==============")
print(similar_scores.sum().sort_values(ascending=False))
print("==================5==============")

```

1.2. File Data mẫu

	action1	action2	action3	romantic1	romantic2	romantic3
user 1	4	5	3		2	1
user 2	5	3	3	2	2	
user 3	1			4	5	4
user 4		2	1	4		3
user 5	1		2	3	3	4

1.3. Giải thích code:

- **import pandas as pd** : chúng ta tiến hành import thư viện Pandas, nó là một thư viện Python cung cấp các cấu trúc dữ liệu nhanh, mạnh mẽ, linh hoạt và mang hàm ý. Chúng em dùng nó để đọc dữ liệu từ file csv.
- **from scipy import sparse** : scipy là một thư viện mã nguồn mở các thuật toán và các công cụ toán học cho Python, được xây dựng trên các đối tượng mảng NumPy tạo thành ngăn xếp NumPy bao gồm các công cụ như Pandas, SymPy và Matplotlib. SciPy cung cấp khá nhiều module tính toán từ đại số tuyến tính, tích phân, vi phân, nội suy đến xử lý ảnh, fourier transform... và cụ thể ở đây ta import gói ma trận sparse và các đoạn chương trình liên quan

- **from sklearn.metrics.pairwise import cosine_similarity** : cosine_similarity được lấy từ sklearn.metrics.pairwise để tính toán độ tương tự giữa 2 vectors
- **ratings=pd.read_csv("toy_dataset.csv",index_col=0):** gán biến ratings với dữ liệu là file toy_dataset.csv bằng việc đọc vào một file .csv bằng cách sử dụng hàm read_csv và được trả về 1 dataframe. i*ndex_col=0* chỉ định chỉ số cột là cột chỉ số (trường hợp chúng ta chọn là cột 0).
- **ratings.fillna(0, inplace=True)** : dùng để chuẩn hóa dữ liệu, hàm này sẽ thay đổi các gí trị NaN bằng giá trị 0;
-
```
def standardize(row):
    new_row = (row - row.mean())/(row.max()-row.min())
    return new_row
df_std = ratings.apply(standardize).T
print(df_std)
```
: function này dùng để chúng ta chuẩn hóa lại dữ liệu vì khi thay giá trị NaN bằng giá trị 0 thì có nghĩa người dùng này thật sự rất không thích sản phẩm đó. Điều này không thực sự tốt vì giá trị '0' tương ứng với mức độ quan tâm thấp nhất. Một giá trị an toàn hơn là 2.5 vì nó là trung bình cộng của 0, mức thấp nhất, và 5, mức cao nhất. Tuy nhiên, giá trị này có hạn chế đối với những users dễ tính hoặc khó tính. Với các users dễ tính, thích tương ứng với 5 sao, không thích có thể ít sao hơn 1 chút, 3 sao chẳng hạn. Việc chọn giá trị 2.5 sẽ khiến cho các items còn lại là quá negative đối với user đó. Điều ngược lại xảy ra với những user khó tính hơn khi chỉ cho 3 sao cho các items họ thích và ít sao hơn cho những items họ không thích. Một giá trị khả dĩ hơn cho việc này là trung bình cộng của các ratings mà user tương ứng đã thực hiện. Việc này sẽ tránh được việc users quá khó tính hoặc dễ tính, tức lúc nào cũng có những items mà một user thích hơn so với những items khác. Chính vì thế ở bước này chúng em chọn cách chuẩn hóa đó là lấy giá trị rating của người dùng trừ cho giá trị trung bình của rating và tiến hành chia cho mức rating trung bình của cả hệ thống (giá trị giữa min và max rating).

- sparse_df = sparse.csr_matrix(df_std.values) : dùng để khai báo sparse matrix
-
```
corrMatrix = pd.DataFrame(cosine_similarity(sparse_df),index=ratings.columns,columns=ratings.columns)
```
: khai báo bảng ghi mới với giá trị là kết quả tính cosine_similarity (góc của 2 vector).
-
```
def get_similar(movie_name,rating):
    similar_score = corrMatrix[movie_name]*(rating-2.5)
    similar_score = similar_score.sort_values(ascending=False)
    #print(type(similar_ratings))
    return similar_score
```
: dùng để tính chỉ số tương tự đối với việc rating 1 film của người dùng. Từ

chính chỉ số rating của người dùng đó với 1 bộ phim so với bảng ghi chứa kết quả cosine_similarity ta đã tính được ở trên.

- `print(similar_scores.sum().sort_values(ascending=False))`: từ những rating chúng ta đã tính toán được độ tương tự của người dùng qua từng rating, từ những kết quả đó chúng ta cộng nó lại để ra kết quả cuối cùng. Điểm càng cao tương đương với mức độ tương đồng gợi ý càng cao.

1.4. Kết quả

```
         action1  action2  action3  romantic1  romantic2  romantic3
user 1       4.0      5.0      3.0        0.0        2.0        1.0
user 2       5.0      3.0      3.0        2.0        2.0        0.0
user 3       1.0      0.0      0.0        4.0        5.0        4.0
user 4       0.0      2.0      1.0        4.0        0.0        3.0
user 5       1.0      0.0      2.0        3.0        3.0        4.0
-----------------------------------------------
           user 1  user 2  user 3    user 4    user 5
action1      0.36    0.56   -0.24 -0.440000 -0.240000
action2      0.60    0.20   -0.40  0.000000 -0.400000
action3      0.40    0.40   -0.60 -0.266667  0.066667
romantic1   -0.65   -0.15    0.35  0.350000  0.100000
romantic2   -0.08   -0.08    0.52 -0.480000  0.120000
romantic3   -0.35   -0.60    0.40  0.150000  0.400000
-----------------------------------------------
            action1   action2   action3  romantic1  romantic2  romantic3
action1    1.000000  0.706689  0.813682  -0.799411  -0.025392  -0.914106
action2    0.706689  1.000000  0.723102  -0.845154  -0.518999  -0.843374
action3    0.813682  0.723102  1.000000  -0.847946  -0.379980  -0.802181
romantic1 -0.799411 -0.845154 -0.847946   1.000000   0.148039   0.723747
romantic2 -0.025392 -0.518999 -0.379980   0.148039   1.000000   0.393939
romantic3 -0.914106 -0.843374 -0.802181   0.723747   0.393939   1.000000
-----------------------------------------------
            action1   action2   action3  romantic1  romantic2  romantic3
action1    1.000000  0.706689  0.813682  -0.799411  -0.025392  -0.914106
action2    0.706689  1.000000  0.723102  -0.845154  -0.518999  -0.843374
action3    0.813682  0.723102  1.000000  -0.847946  -0.379980  -0.802181
romantic1 -0.799411 -0.845154 -0.847946   1.000000   0.148039   0.723747
romantic2 -0.025392 -0.518999 -0.379980   0.148039   1.000000   0.393939
romantic3 -0.914106 -0.843374 -0.802181   0.723747   0.393939   1.000000
-----------------------------------------------
-----------------------------------------------
    action1   action2   action3  romantic1  romantic2  romantic3
0  2.500000  1.766722  2.034204  -1.998527  -0.063480  -2.285265
1  0.038088  0.778499  0.569970  -0.222059  -1.500000  -0.590909
2  1.371159  1.265061  1.203271  -1.085620  -0.590909  -1.500000
-----------------------------------------------
action1      3.909247
action2      3.810282
action3      3.807445
romantic2   -2.154389
romantic1   -3.306206
romantic3   -4.376174
dtype: float64
```

2. Công việc tuần sau:

- Demo dựa theo Content Based