Ngoài cách lọc theo cộng tác Collaborative Filtering thì nhóm chúng em còn kết hợp song song với việc lọc theo nội dung Content-based filtering để tối ưu bài toán Recommend System.

Đầu tiên chúng em sẽ tiền xử lý dữ liệu hai File Ratings.csv và File Movies.csv. Với File Ratings.csv gồm bốn cột: UserId, MovieId, Rating, timestamp thì sẽ đọc vào DataFrame Ratingdata và xoá bỏ cột không cần thiết là timestamp.

Còn File thứ 2 gồm 3 cột: MovieId, title, genres thì chúng em sẽ xử lý như sau. Đầu tiên tương tự như trên, đọc file vào dataFrame moviedata. Từ trường title gồm 2 thông tin: movie name + year nên chúng em sẽ tách movieName ra thành 1 trường riêng. Tiếp theo từ cột Genres ta sẽ tạo ra 1 list các genre, với mỗi genre sẽ tương ứng với một trường riêng mới. Nếu dataFrame này có genres chứa 1 genra A bất thì trường A bằng 1 tại vị trí tương ứng với movie đó, ngược lại thì bằng 0. Cuối cùng in ra list các genres và moviedata trước và sau khi tiền xử lý.

Tiếp theo để training lọc theo nội dung chúng em đã chọn model RandomForest dạng Regression. Từ 2 dataFrame đã tiền xử lý, chúng em lọc ra trường UserId của Ratingdata và 19 trường genre của Moviedata làm Input Bộ dữ liệu thì được chia một cách ngẫu nhiên thành hai phần 80% Train- 20% Test.

Thuật toán Random Forest sẽ tạo ra K tree bằng cách sau. Đầu tiên sẽ dùng BootStrap Sampling để tạo ra Traning Set tương ứng với Cây thứ i. Với mỗi node cả cây sẽ chọn một tập con các Attributes và split theo tập con đó. Hàm dùng để đo độ Split là MSE. Cây sẽ được expand tối đa và không cần cắt tỉa. Giá trị output sẽ được xác định là giá trị trung bình dự đoán của tất cả các cây.

Cuối cùng chúng em dùng hàm đánh giá là Root Mean Squared Error để so sánh kết quả dự đoán với giá trị thực

Nhóm đã thử với số lượng cây khác nhau để tìm ra giá trị tốt nhất.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| K | 10 | 50 | 100 | 300 |
| RMSE | 0,976 | 0,98 | 0,974 | 0,977 |

Thuật toán RadomForest cho ra kết quả khá tốt tăng số cây lên.