BÁO CÁO TIẾN ĐỘ TIỂU LUẬN CHUYÊN NGÀNH

(TUẦN 3)

Đề tài: Tìm hiểu các thuật toán Recommendation

Link github: Recommendation

Thành viên nhóm:

Trần Nguyên Thái Bảo 19133010

Đinh Quốc Hùng 19133025

❖ Kết quả đạt được

4. Demo Content-Based Recommendations sử dụng thuật toán TF-IDF

4.3. Xây dựng demo

4.3.3. Học mô hình cho từng người dùng

- Sau khi đã xây dựng items profile cho mỗi movie dựa trên 19 thể loại, lấy ví dụ feature vector cho mỗi bộ phim theo bảng ở dưới:

	user 1	user 2	user 3	user 4	user 5	Feature vector
MV 1	**5**	**?**	**1**	**5**	**2**	**f1=[g1,...,g19]**
MV 2	**2**	**4**	**?**	**?**	**?**	**f2=[g1,...,g19]**
MV 3	**1**	**4**	**2**	**?**	**?**	**f3=[g1,...,g19]**
MV 4	**?**	**?**	**0**	**1**	**5**	**f4=[g1,...,g19]**
Mô hình cho user	**θ_1**	**θ_2**	**θ_3**	**θ_4**	**θ_5**	

=> Với mỗi *user*, ta cần tìm một mô hình θ tương ứng sao cho mô hình thu được là *tốt nhất*. Và bài toán này được coi là bài toán hồi quy (Regression), cụ thể trong phần này nhóm sẽ sử dụng Linear Regression với L2 Regularization (tham khảo)

$$\mathcal{L}_n = \frac{1}{2s_n}||\hat{\mathbf{X}}_n\mathbf{w}_n + b_n\mathbf{e}_n - \hat{\mathbf{y}}_n||_2^2 + \frac{\lambda}{2s_n}||\mathbf{w}_n||_2^2$$

- Sử dụng Ridge trong sklearn.linear_model (tham khảo biến và thuộc tính của Ridge) để tìm ra cặp nghiệm Wn và bn

Học mô hình cho mỗi user với L2 Regularization

```
from sklearn.linear_model import Ridge
from sklearn import linear_model

d = tfidf.shape[1]
W = np.zeros((d, n_users))
b = np.zeros((1, n_users))
for n in range(n_users):
    ids, scores = get_items_rated_by_user(rate_train, n)
    clf = Ridge(alpha= 0.01, fit_intercept  = True)
    Xhat = tfidf[ids, :]

    clf.fit(Xhat, scores)
    W[:, n] = clf.coef_  # ndarray of shape (n_features,) or (n_targets, n_features)
    b[0, n] = clf.intercept_ # float or ndarray of shape (n_targets,)
```

✓ 1.6s

- Sử dụng cặp nghiệm W b vừa tìm được để dự đoán ma trận

```
# Use W and b to finish utility matrix
Yhat = tfidf.dot(W) + b
```

✓ 0.1s

- So sánh kết quả dự đoán được với kết quả trên tập test
  Ví dụ với UserId = 7

```
# Test with userId = 7
n = 7
#np.set_printoptions(precision=2) # 2 digits after .
ids, scores = get_items_rated_by_user(rate_test, n)
compareResult = {'MovieId':ids,'True rate':scores,'Predicted':Yhat[ids,n]}
display(pd.DataFrame(compareResult))
```

✓ 0.2s

	MovieId	True rate	Predicted
0	21	5	4.331464
1	49	5	4.612072
2	78	4	3.650662
3	88	4	3.847020
4	181	5	4.076319
5	293	3	2.122395
6	337	4	2.122395
7	384	1	3.753040
8	456	1	2.665171
9	549	3	3.650662

4.4. Đánh giá mô hình sử dụng Root Mean Squared Error (RMSE)

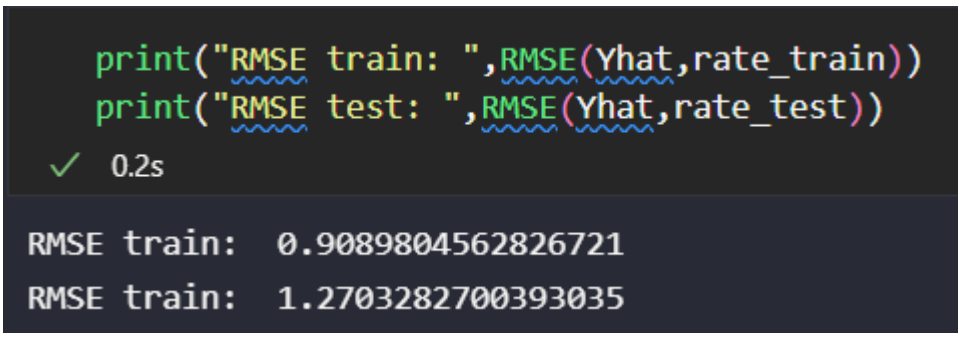
$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- Tạo hàm để tính toán RMSE trên từng tập data

```
# Create function to calculate RMSE
def RMSE(Yhat, data):
    se = 0
    cnt = 0
    for n in range(n_users):
        ids, scores_truth = get_items_rated_by_user(data, n)
        scores_pred = Yhat[ids, n]
        e = scores_truth - scores_pred
        se += (e*e).sum(axis = 0)
        cnt += e.size
    return np.sqrt(se/cnt)
```

✓ 0.4s

- Kết quả



- Nhận xét:
  - RMSE) là một biện pháp thường được sử dụng trong những khác biệt giữa các giá trị được dự đoán bởi một mô hình hay một ước lượng và các giá trị quan sát được.
  - Kết quả RMSE trên tập test có sai số cao hơn tập train, nhìn chung thì cả hai kết quả trên train và test đề khá cao do đó nhận xét rằng kết quả của mô hình chưa thực sự tốt

❖ Kế hoạch cho tuần tiếp theo:

➔ Tìm hiểu về kỹ thuật Collaborative Filtering trong Recommendation

➔ Xác định dataset và đề xuất demo