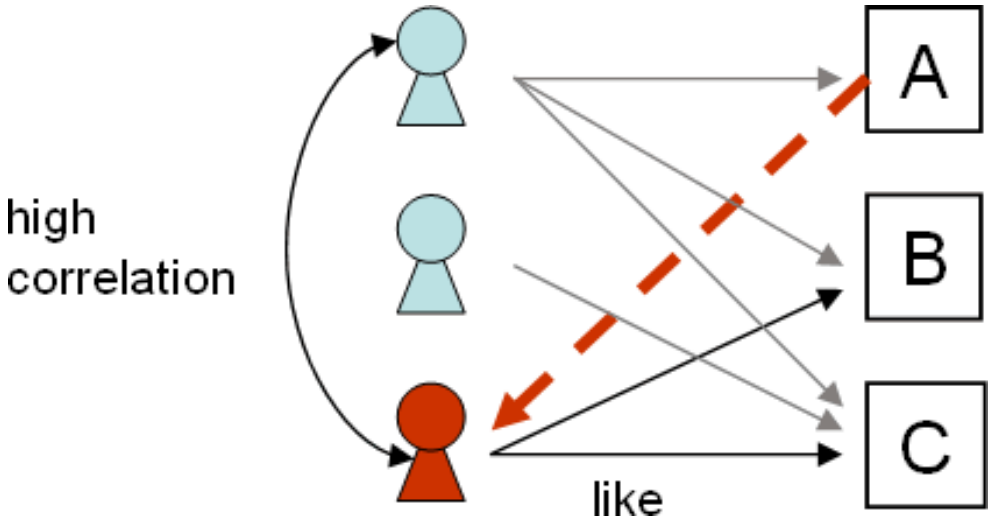
SUNG KYUN KWAN UNIVERSITY

Bi-Weekly Report of (Team #13) Week #11

What were the goals for the last 2 weeks?

[Recommendation System]

- will apply an item collaborative filter with a rating function to reflect the taste of the user
  - and will create rating data (like, dislike, NA)
  - after rating the 10 recommended users above, there will be users(A, B, C) with high ratings. Re-recommend other users(A) selected by the user who gave that user(A) a high rating and exclude users with low ratings.



[Chatting translation]
The goal is to convert the other person's chat into my fluent language. After analyzing the languages supported by Google translation, this is reflected in the system. Because translated messages vary depending on the fluent language, the messages are not stored in the database and are translated in real-time.

[UI improvement]
Improve the user interface and add validation to the sign-up and login screens.

What goals were accomplished this week?

[Recommendation System]

- Create user rating data(point from 1 to 3) to reflect user tastes
  - Each standard is when the gender is the same, when the gender is different, when the age group is the same, when the foreign language level is high and just average rating(point 2).
  - Each user has different standards, reflecting the different tastes of each user (according to user ID mod 5).
  - Only users from the content based recommendation are rated.
  - The created ratings.csv consists of the corresponding user ID, target user ID, and rating.

```python
def generate_rating(user1, user2):
    # 성별이 같으면 3점, 다르면 1점
    if user1['성별'] == user2['성별']:
        criterion1 = 3
    else:
        criterion1 = 1

    # 성별이 같으면 1점, 다르면 3점
    if user1['성별'] == user2['성별']:
        criterion2 = 1
    else:
        criterion2 = 3

    # 나이대가 같으면 3점, 다르면 1점
    if user1['연령대'] == user2['연령대']:
        criterion3 = 3
    else:
        criterion3 = 1

    # 수준이 상이면 3점, 하면 1점
    if user2['외국어 수준'] == '상':
        criterion4 = 3
    elif user2['외국어 수준'] == '중':
        criterion4 = 2
    else:
        criterion4 = 1

    # 기준5: 2점
    criterion5 = 2
```

```python
    # user1의 id에 따라 기준 선택
    user_id_mod = user1['User_ID'] % 5
    if user_id_mod == 0:
        criterion = criterion1
    elif user_id_mod == 1:
        criterion = criterion2
    elif user_id_mod == 2:
        criterion = criterion3
    elif user_id_mod == 3:
        criterion = criterion4
    else:
        criterion = criterion5

    # 평점 부여
    if criterion == 1:
        rating = 1
    elif criterion == 2:
        rating = 2
    else:
        rating = 3

    return rating
```

- Apply an item collaborative filter with a ratings.csv
  - Recommend 10 additional users, excluding other users already rated by the user

$$\underbrace{N}\left\{\overbrace{\boxed{A}}^{M}\right. = \underbrace{N}\left\{\overbrace{\boxed{U}}^{N}\overbrace{\boxed{\Sigma}}^{M}\overbrace{\boxed{V^T}}^{M}\right\}\underbrace{M}$$

-

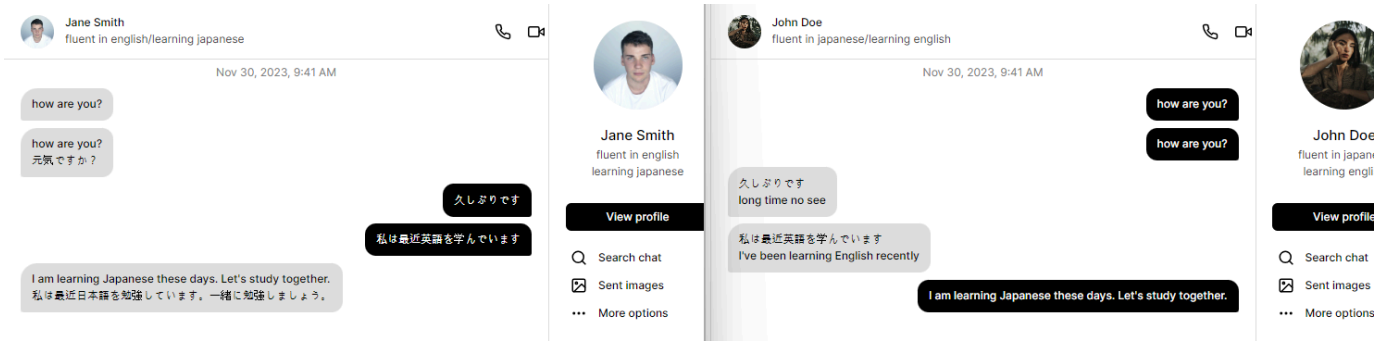
- Collaborative filtering was implemented using singular value decomposition(SVD). The rating matrix given to each user(A) is decomposed into a user matrix(U), characteristic matrix(Σ), and target user matrix(T), and latent factors in the middle matrix are extracted to recommend the top 10 users. These latent factors represent the hidden characteristics of users and target users.

```python
def collaborative_recommend(user_id, train):
    train = match_lang(user_id,train)
    print(train[train['User_ID']==user_id])
    print('='*30)
    for target_user in tqdm(train['User_ID'].values):
        #본인 제외
        if target_user != user_id:
            #이미 평가한 user제외해서 봅음
            compare = ratings[ratings['User_ID'] == user_id]
            if not target_user in compare['Target_User_ID'].values:
                pred = svd.predict(user_id, target_user)  # 사용자 id 1에 대한 각 사용자의 평점 예측
                predictions.append((target_user, pred.est))  # 사용자 id와 예측 평점을 튜플로 저장
    # 예측된 평점을 기준으로 사용자 정렬하여 상위 10개 선택
    top_users = sorted(predictions, key=lambda x: x[1], reverse=True)[:10]
    return top_users
```

- Users recommended by collaborative filtering to user ID 30000

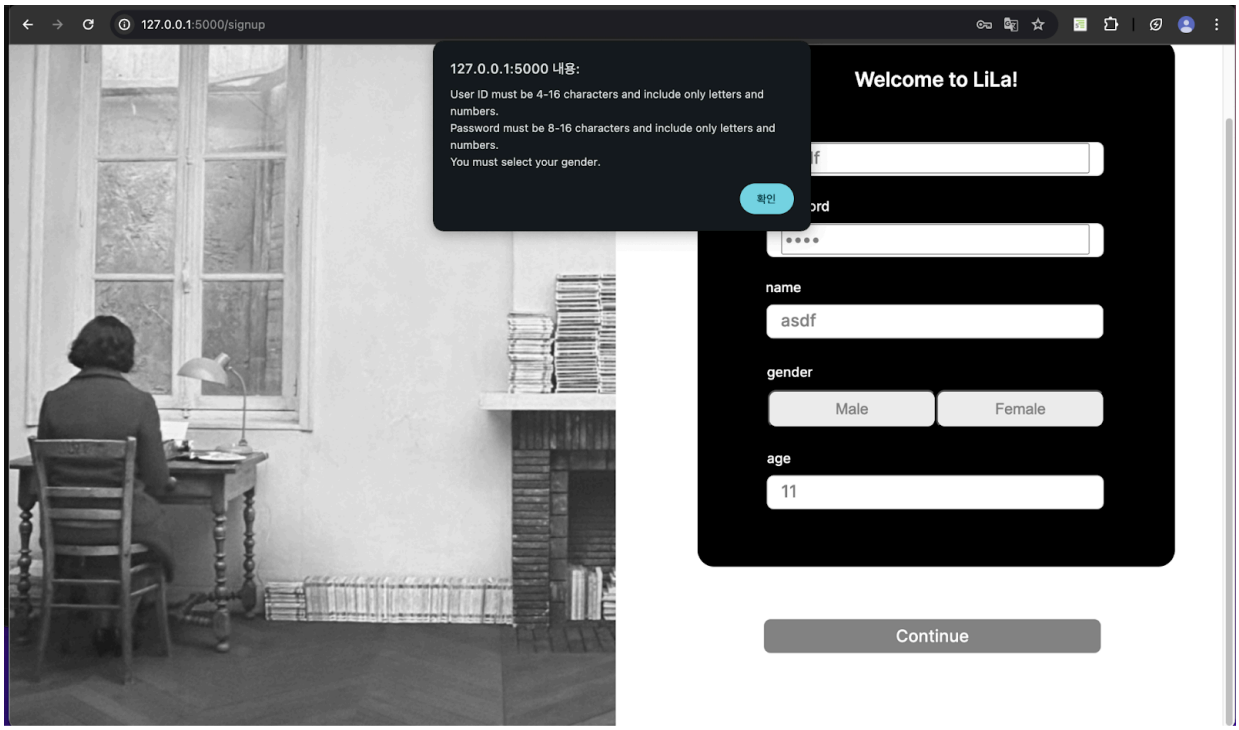
```
      User_ID    이름 성별  연령대                                          자기 소개   모국어 외국어 외국어 수준
29999    30000  Sylus  남  20대  나는 토목 엔지니어이다. 나의 MBTI는 ENFP이다. 나는 팝 음악 감상을(를) ...  스페인어  영어      상
==============================
100%|██████████| 1192/1192 [00:01<00:00, 832.20it/s]
User ID: 23346, Name: Micheala, Rate: 2.201197441420132
User ID: 19603, Name: Kynedi, Rate: 2.18884095920897
User ID: 1609, Name: Alilyana, Rate: 2.187522729062833
User ID: 32808, Name: Yazlyn, Rate: 2.1865528845535094
User ID: 21388, Name: Machelle, Rate: 2.186252075589552
User ID: 45822, Name: Semisi, Rate: 2.186118958607957
User ID: 16426, Name: Juliani, Rate: 2.1855679895513833
User ID: 27550, Name: Rudi, Rate: 2.185347182131777
User ID: 24164, Name: Najae, Rate: 2.1852115003111754
User ID: 23637, Name: Mirya, Rate: 2.1843492196776255
```

[Chatting translation]



If the chat language is included in the supported languages, it will be translated into the other person's native language. The supported languages are [english, chinese, arabic, spanish, french, korean, japanese].

[UI improvement]



When entering the password into the password input field, the text was originally exposed as is. This part was hidden. In addition, through various validation checks, if incorrect input values are entered, the alert function alerts you to what was wrong when signing up.

```
function validateForm(id, pw, name, age, gender) {

    // 유저 아이디 검사
    if (!id) {
        errors.push('You must write down your user ID.');
    } else if (!/^[a-zA-Z0-9]{4,16}$/.test(id)) {
        errors.push('User ID must be 4-16 characters and include only letters and numbers.');
    }

    // 비밀번호 검사
    if (!pw) {
        errors.push('You must write down your password.');
    } else if (!/^[a-zA-Z0-9]{8,16}$/.test(pw)) {
        errors.push('Password must be 8-16 characters and include only letters and numbers.');
    }

    // 이름 검사
    if (!name) {
        errors.push('You must write down your name.');
    } else if (!/^[a-zA-Z]+$/.test(name)) {
        errors.push('Name must include only letters.');
    }

    // 나이 검사
    if (!age) {
        errors.push('You must write down your age.');
    } else if (!/^\d+$/.test(age)) {
        errors.push('Age must be numeric.');
    }

    // 성별 검사
    if (!gender) {
        errors.push('You must select your gender.');
    }

    if (errors.length > 0) {
        alert(errors.join('\n'));
        return false;
    }
    return true;
}
```

Reflect critically on any goals not accomplished.

- We tried to add recommendations based on MBTI, but could not do so for the following reasons.
    - Since the hobbies entered by each user are in a free format, it is highly likely that MBTI will not be included.
    - It is unscientific to recommend people based on MBTI compatibility.

What are the goals for the next two weeks?

- Reflect the recommendation system on the server
- Translate user input in Korean (by googletrans)
- Deploy environment analysis
- UI improvement(make recommendation button, refresh button, and give tasks to each button)

How many hours were spent on each goal noted above?

Recommendation system
- 20 hours per two weeks

Backend
- 15 hours per two weeks

Frontend
- 10 hours per two weeks