

# Surprise 라이브러리 비교 - 20212033 김종훈

## 주요 변경점

### 1) 유사도 계산 및 처리

- 중간고사

```
# 1단계: Pearson 상관계수 계산
user_similarity_org = train_user_movie_rates.T.corr(method='pearson')

# 2단계: 음수 유사도 제거 ✗
user_similarity = user_similarity_org[user_similarity_org >= 0.0].fillna(0)
```

- Surprise 라이브러리 이용

```
sim_options = {
    'name': 'pearson',
    'user_based': True
}
algo = KNNWithMeans(k=10, sim_options=sim_options)
```

### 2) 예측 함수

- 중간고사
  - 직접 bias기반 예측 평점 계산 수행

```
def CF_knn_bias(user_id, movie_id, neighbor_size=10):
    if user_id not in rating_bias.index:
        prediction = 0
        return prediction

    if movie_id in rating_bias:
        sim_scores = user_similarity[user_id].copy()
        movie_ratings = rating_bias[movie_id].copy()
        # ... 복잡한 로직 ...
        prediction = np.dot(sim_scores, movie_ratings) / sim_scores.sum()
        prediction = prediction + rating_mean[user_id]
    else:
        prediction = rating_mean[user_id]

    return prediction
```

- surprise 라이브러리 이용

```
pred = algo.predict(user, movie)
predicted_rating = pred.est
```

- 간단하게 사용 가능

### 3) RMSE 계산

- 중간고사
  - RMSE 수동으로 코드 계산

```
# 수동으로 예측
y_pred = np.array([CF_knn_bias(user, movie, neighbor_size=10)
                    for (user, movie) in id_pairs])

def RMSE(y_true, y_pred):
    return np.sqrt(np.mean((np.array(y_true) - np.array(y_pred))**2))

print(RMSE(y_true, y_pred))
```

- surprise 이용

```
predictions = [algo.predict(uid, iid, r_ui=true_r)
                for uid, iid, true_r in zip(x_test['user'], x_test['movie'], x_test['rate'])]

rmse = accuracy.rmse(predictions, verbose=False)
```

## 주요 결과 비교

### 1) RMSE

- 중간고사

```
print(RMSE(y_true, y_pred))
```

- 7.267291364193568

- Surprise 활용

```
rmse = accuracy.rmse(predictions, verbose=False)
rmse
```

- 1.874818543984475

- 이유 추정

- 중간고사때 음수 상관관계 사용 X
- Surprise의 KNNWithMeans는 음수 상관관계까지 모두 학습에 활용됨
- 이에따라 Surprise를 활용한 모델이 더 정확한 평점을 예측 했을 것

### 2) Precision@10, Recall@10

- 중간고사

```
precision_at_10, recall_at_10 = precision_recall_at_k(true_user2items, pred_user2items, k=10)
print(f"Precision@10: {precision_at_10:.4f}")
print(f"Recall@10: {recall_at_10:.4f}")
```

- Precision@10: 0.4250
- Recall@10: 0.1417

- surprise 이용

```
precision_at_10, recall_at_10 = precision_recall_at_k(true_user2items, pred_user2items, k=10)

print(f"Precision@10: {precision_at_10:.4f}")
print(f"Recall@10: {recall_at_10:.4f}")
```

- Precision@10: 0.0500