

```

# 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

- 중간고사땐 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

- Recall@10: 0.0167
- 이유 추정
  - 중간고사 코드는 양의 상관관계만 사용 → 사용자가 뭘 좋아했는지만 학습했다
  - surprise 코드는 양의 상관관계 + 음의 상관관계 → 사용자가 뭘 좋아하고, 뭘 싫어하는지를 학습함
  - surprise는 사용자가 매길 모든 영화의 평점을 정확히 맞추는데 집중 (RMSE낮음)
    - 다만, 사용자의 취향에 의외로 맞을 순 있지만, 음의 상관관계에 따라 예측 평점이 낮게 나와서, 숨겨진 보석같은 영화가 추천이 안될 수 있음
  - 중간고사 코드는 사용자가 싫어할만한 영화에 대한 예측은 실패할 수 있지만, 오히려 사용자의 취향에 맞는 새로운 영화를 발견할 수 있음
  - 이에따라 중간고사 코드가, 더 Precision과 Recall이 높게 나왔을 가능성 높음