Artificial Intelligence Experiment

7주차 결과 보고서

전공: 컴퓨터공학과
학년: 3학년
학번: 20221548
이름: 김호정

1 가상의 유저 944와 945가 부여한 별점에 대한 정보를 첨부하고, 해당 별점을 부여한 이유를 취향의 측면에서 작성한다.

user 944는 'Horror','Thriller' 영화에 높은 평점을, 'Children's', 'Musical'영화에 낮은 평점을 준 user이다.

User 944's ratings in train

	movie_id	rating	movie_title	release_date	genres
0	2	5	GoldenEye (1995)	01-Jan-1995	Action, Adventure, Thriller
1	3	5	Four Rooms (1995)	01-Jan-1995	Thriller
2	5	5	Copycat (1995)	01-Jan-1995	Crime, Drama, Thriller
3	11	5	Seven (Se7en) (1995)	01-Jan-1995	Crime, Thriller
4	12	5	Usual Suspects, The (1995)	14-Aug-1995	Crime, Thriller
5	17	5	From Dusk Till Dawn (1996)	05-Feb-1996	Action, Comedy, Crime, Horror, Thriller
6	23	5	Taxi Driver (1976)	16-Feb-1996	Drama, Thriller
7	84	5	Robert A. Heinlein's The Puppet Masters (1994)	01-Jan-1994	Horror, Sci-Fi
8	183	5	Alien (1979)	01-Jan-1979	Action, Horror, Sci-Fi, Thriller
9	185	5	Psycho (1960)	01-Jan-1960	Horror, Romance, Thriller
10	200	5	Shining, The (1980)	01-Jan-1980	Horror
11	1	1	Toy Story (1995)	01-Jan-1995	Animation, Children's, Comedy
12	8	1	Babe (1995)	01-Jan-1995	Children's, Comedy, Drama
13	35	1	Free Willy 2: The Adventure Home (1995)	01-Jan-1995	Adventure, Children's, Drama
14	63	1	Santa Clause, The (1994)	01-Jan-1994	Children's, Comedy
15	71	1	Lion King, The (1994)	01-Jan-1994	Animation, Children's, Musical
16	78	1	Free Willy (1993)	01-Jan-1993	Adventure, Children's, Drama
17	91	1	Nightmare Before Christmas, The (1993)	01-Jan-1993	Children's, Comedy, Musical
18	94	1	Home Alone (1990)	01-Jan-1990	Children's, Comedy
19	95	1	Aladdin (1992)	01-Jan-1992	Animation, Children's, Comedy, Musical
20	99	1	Snow White and the Seven Dwarfs (1937)	01-Jan-1937	Animation, Children's, Musical
21	103	1	All Dogs Go to Heaven 2 (1996)	29-Mar-1996	Animation, Children's, Musical
22	132	1	Wizard of Oz, The (1939)	01-Jan-1939	Adventure, Children's, Drama, Musical
23	142	1	Bedknobs and Broomsticks (1971)	01-Jan-1971	Adventure, Children's, Musical

user 945는 'Animation', 'Fantasy' 영화에 높은 평점을, 'War','Crime', 'Western'영화에 낮은 평점을 준 user이다.

User 945's ratings in train

	movie_id	rating	movie_title	release_date	genres
0	1	5	Toy Story (1995)	01-Jan-1995	Animation, Children's, Comedy
1	71	5	Lion King, The (1994)	01-Jan-1994	Animation, Children's, Musical
2	72	5	Mask, The (1994)	01-Jan-1994	Comedy, Crime, Fantasy
3	95	5	Aladdin (1992)	01-Jan-1992	Animation, Children's, Comedy, Musical
4	99	5	Snow White and the Seven Dwarfs (1937)	01-Jan-1937	Animation, Children's, Musical
5	101	5	Heavy Metal (1981)	08-Mar-1981	Action, Adventure, Animation, Horror, Sci-Fi
6	102	5	Aristocats, The (1970)	01-Jan-1970	Animation, Children's
7	103	5	All Dogs Go to Heaven 2 (1996)	29-Mar-1996	Animation, Children's, Musical
8	114	5	Wallace & Gromit: The Best of Aardman Animatio...	05-Apr-1996	Animation
9	141	5	20,000 Leagues Under the Sea (1954)	01-Jan-1954	Adventure, Children's, Fantasy, Sci-Fi
10	169	5	Wrong Trousers, The (1993)	01-Jan-1993	Animation, Comedy
11	189	5	Grand Day Out, A (1992)	01-Jan-1992	Animation, Comedy
12	240	5	Beavis and Butt-head Do America (1996)	20-Dec-1996	Animation, Comedy
13	308	5	FairyTale: A True Story (1997)	01-Jan-1997	Children's, Drama, Fantasy
14	404	5	Pinocchio (1940)	01-Jan-1940	Animation, Children's
15	411	5	Nutty Professor, The (1996)	28-Jun-1996	Comedy, Fantasy, Romance, Sci-Fi
16	423	5	E.T. the Extra-Terrestrial (1982)	01-Jan-1982	Children's, Drama, Fantasy, Sci-Fi
17	548	5	NeverEnding Story III, The (1994)	02-Feb-1996	Children's, Fantasy
18	5	1	Copycat (1995)	01-Jan-1995	Crime, Drama, Thriller
19	10	1	Richard III (1995)	22-Jan-1996	Drama, War
20	11	1	Seven (Se7en) (1995)	01-Jan-1995	Crime, Thriller
21	12	1	Usual Suspects, The (1995)	14-Aug-1995	Crime, Thriller
22	17	1	From Dusk Till Dawn (1996)	05-Feb-1996	Action, Comedy, Crime, Horror, Thriller
23	22	1	Braveheart (1995)	16-Feb-1996	Action, Drama, War
24	29	1	Batman Forever (1995)	01-Jan-1995	Action, Adventure, Comedy, Crime
25	31	1	Crimson Tide (1995)	01-Jan-1995	Drama, Thriller, War
26	50	1	Star Wars (1977)	01-Jan-1977	Action, Adventure, Romance, Sci-Fi, War
27	55	1	Professional, The (1994)	01-Jan-1994	Crime, Drama, Romance, Thriller
28	80	1	Hot Shots! Part Deux (1993)	01-Jan-1993	Action, Comedy, War
29	97	1	Dances with Wolves (1990)	01-Jan-1990	Adventure, Drama, Western
30	110	1	Operation Dumbo Drop (1995)	01-Jan-1995	Action, Adventure, Comedy, War
31	177	1	Good, The Bad and The Ugly, The (1966)	01-Jan-1966	Action, Western
32	203	1	Unforgiven (1992)	01-Jan-1992	Western
33	232	1	Young Guns (1988)	01-Jan-1988	Action, Comedy, Western
34	435	1	Butch Cassidy and the Sundance Kid (1969)	01-Jan-1969	Action, Comedy, Western

2 실습에서 사용한 코드를 첨부하고, 코드를 설명한다.

```
import pyspark
import pandas as pd
```

```
from pyspark.sql import SparkSession
from pyspark.ml.recommendation import ALS
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder

# Spark 세션 생성
spark = SparkSession.builder \
    .appName("MovieLens Recommendation System with CV") \
    .getOrCreate()

# 데이터 로드
train = spark.read.csv("ml-100k/u1.base", header=None, inferSchema=True, sep="\t")
test = spark.read.csv("ml-100k/u1.test", header=None, inferSchema=True, sep="\t")

train = train.toDF("user_id", "movie_id", "rating", "timestamp")
test = test.toDF("user_id", "movie_id", "rating", "timestamp")
```

아래 코드애서는 u.item 파일을 열어 영화 정보에 대한 데이터프레임을 생성한다.

```
# u.item 파일 경로를 지정
file_path = 'ml-100k/u.item'

# 데이터를 불러오기
column_names = ['movie_id', 'movie_title', 'release_date', 'video_release_date', 'IMDb_URL',
'unknown', 'Action', 'Adventure', 'Animation', 'Children\'s', 'Comedy', 'Crime', 'Documentary',
'Drama', 'Fantasy', 'Film-Noir', 'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi',
'Thriller', 'War', 'Western']
df = pd.read_csv(file_path, sep='|', names=column_names, encoding='latin-1')

# 필요한 열만 선택
selected_columns = df[['movie_id', 'movie_title', 'release_date']]

# 액션 장르를 포함한 모든 영화 데이터프레임 저장
movies_df = df[['movie_id', 'movie_title', 'release_date', 'Action', 'Adventure', 'Animation',
'Children\'s', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir', 'Horror',
'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War', 'Western']]

display(movies_df)
```

	movie_id	movie_title	release_date	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	...	Fantasy	Film-Noir	Horror	Musical	Mystery	Ror
0	1	Toy Story (1995)	01-Jan-1995	0	0	1	1	1	0	0	...	0	0	0	0	0	
1	2	GoldenEye (1995)	01-Jan-1995	1	1	0	0	0	0	0	...	0	0	0	0	0	
2	3	Four Rooms (1995)	01-Jan-1995	0	0	0	0	0	0	0	...	0	0	0	0	0	
3	4	Get Shorty (1995)	01-Jan-1995	1	0	0	0	1	0	0	...	0	0	0	0	0	
4	5	Copycat (1995)	01-Jan-1995	0	0	0	0	0	1	0	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1677	1678	Mat' i syn (1997)	06-Feb-1998	0	0	0	0	0	0	0	...	0	0	0	0	0	
1678	1679	B. Monkey (1998)	06-Feb-1998	0	0	0	0	0	0	0	...	0	0	0	0	0	
1679	1680	Sliding Doors (1998)	01-Jan-1998	0	0	0	0	0	0	0	...	0	0	0	0	0	
1680	1681	You So Crazy (1994)	01-Jan-1994	0	0	0	0	1	0	0	...	0	0	0	0	0	
1681	1682	Scream of Stone (Schrei aus Stein) (1991)	08-Mar-1996	0	0	0	0	0	0	0	...	0	0	0	0	0	

1682 rows × 21 columns

아래 코드의 get_genres 함수를 사용하면 원 핫 인코딩 방식으로 저장된 영화 장르에 대한 정보를 처리하여 데이터 프레임으로 나타낼 때 영화 장르 값이 한눈에 표시되게 할 수 있다.

```
def get_genres(row):
    genres = []
    for genre in column_names[6:]:
        if row[genre] == 1:
            genres.append(genre)
    return ', '.join(genres)
```

아래 코드는 ALS 알고리즘을 활용해 5 fold crossvalidation을 통해 RMSE를 최소화하는 최적의 모델을 선택하고, 테스트 데이터에 대한 RMSE(root mean square error)를 계산하는 과정이다.

```
# ALS 모델 생성
als = ALS(
    userCol="user_id",
    itemCol="movie_id",
    ratingCol="rating",
    coldStartStrategy="drop"
)

# 파라미터 그리드 설정
paramGrid = ParamGridBuilder() \
    .addGrid(als.maxIter, [5, 10, 15]) \
    .addGrid(als.regParam, [0.01, 0.1, 0.5]) \
    .build()

# 평가기 설정
evaluator = RegressionEvaluator(
    metricName="rmse",
    labelCol="rating",
    predictionCol="prediction"
)
```

```
# 교차 검증기 설정
crossval = CrossValidator(
    estimator=als,
    estimatorParamMaps=paramGrid,
    evaluator=evaluator,
    numFolds=5 # 5겹 교차 검증
)
# 교차 검증 모델 학습
cvModel = crossval.fit(train)

# 최적의 모델 선택
bestModel = cvModel.bestModel

# 최적 모델의 파라미터 출력
print(f"Best maxIter: {bestModel._java_obj.parent().getMaxIter()}")
print(f"Best regParam: {bestModel._java_obj.parent().getRegParam()}")

# 테스트 데이터에 대한 RMSE 평가
predictions = bestModel.transform(test)
rmse = evaluator.evaluate(predictions)
print(f"Root-mean-square error = {rmse}")
```

결과는 다음과 같다.

Best maxIter: 15
Best regParam: 0.1
Root-mean-square error = 0.933501740952482

아래 코드에서는 모든 사용자에게 추천할 10개의 영화를 보여준다.

```
# 사용자에게 추천할 아이템 추출
userRecs = bestModel.recommendForAllUsers(10)
userRecs.show(5, truncate=False)
```

이제 사용자 944, 945처럼 특정 사용자에게 10개의 영화를 추천하고 보기 좋게 나타내보자.

```
def print_recommendations(userId):
    print(f'User {userId}\'s recommendation')
    user_subset = train.filter(train.user_id == userId).select("user_id").distinct()

    # 해당 사용자에 대한 추천 추출
    user_recs = bestModel.recommendForUserSubset(user_subset, 10)
    # user_recs.show(truncate=False)
    df = user_recs.select('recommendations').toPandas()

    # recommendations 컬럼의 데이터를 확장
    expanded_rows = []
    for row in df.itertuples(index=False):
        expanded_rows.extend(row.recommendations)

    # 새로운 DataFrame 생성
    expanded_df = pd.DataFrame(expanded_rows, columns=['movie_id', 'rating'])

    # 영화 정보와 추천 결과 병합
    merged_df = pd.merge(expanded_df, movies_df, on='movie_id')

    # 장르 컬럼 이름을 합친 새로운 컬럼 추가
    merged_df['genres'] = merged_df.apply(get_genres, axis=1)
```

```
    # 필요 없는 장르 컬럼 삭제
    result_df = merged_df[['movie_id', 'rating', 'movie_title', 'release_date', 'genres']]

    # 결과 출력
    display(result_df)

userId = 944
print_recommendations(userId)
```

User 944's recommendation

	movie_id	rating	movie_title	release_date	genres
0	695	6.015628	Kicking and Screaming (1995)	01-Jan-1995	Comedy, Drama
1	156	5.760349	Reservoir Dogs (1992)	01-Jan-1992	Crime, Thriller
2	763	5.243705	Happy Gilmore (1996)	16-Feb-1996	Comedy
3	56	5.200497	Pulp Fiction (1994)	01-Jan-1994	Crime, Drama
4	899	5.105593	Winter Guest, The (1997)	01-Jan-1997	Drama
5	943	5.096783	Killing Zoe (1994)	01-Jan-1994	Thriller
6	410	5.055227	Kingpin (1996)	12-Jul-1996	Comedy
7	203	5.009650	Unforgiven (1992)	01-Jan-1992	Western
8	42	5.008235	Clerks (1994)	01-Jan-1994	Comedy
9	1010	4.996832	Basquiat (1996)	16-Aug-1996	Drama

```
userId = 945
print_recommendations(userId)
```

User 945's recommendation

	movie_id	rating	movie_title	release_date	genres
0	1233	6.034416	Nénette et Boni (1996)	01-Jan-1996	Drama
1	1138	5.828148	Best Men (1997)	01-Sep-1997	Action, Comedy, Crime, Drama
2	1609	5.531354	B*A*P*S (1997)	28-Mar-1997	Comedy
3	6	5.514738	Shanghai Triad (Yao a yao yao dao waipo qiao) ...	01-Jan-1995	Drama
4	1066	5.421926	Balto (1995)	01-Jan-1995	Animation, Children's
5	1431	5.309130	Legal Deceit (1997)	01-Jan-1997	Thriller
6	1643	5.306767	Angel Baby (1995)	10-Jan-1997	Drama
7	261	5.183487	Air Bud (1997)	01-Aug-1997	Children's, Comedy
8	962	5.166711	Ruby in Paradise (1993)	01-Jan-1993	Drama
9	1147	5.121228	My Family (1995)	01-Jan-1995	Drama

3 학습 후 총 2명의 가상 유저의 추천 결과를 분석한다.

추천 결과와 직접 별점을 부여한 영화와 점수에 대한 상관관계에 대해서는 필수적으로 작성해야 함.

다시 한 번 train에 넣은 user rating을 보면 user 944는 'Horror','Thriller' 영화에 높은 평점을, 'Children's', 'Musical' 영화에 낮은 평점을 준 user이다. user 945는 'Animation', 'Fantasy' 영화에 높은 평점을, 'War','Crime', 'Western' 영화에 낮은 평점을 준 user이다.

User 944는 Children's, Musical 장르와는 거리가 먼 영화들이 추천된다는 점에서 낮은 rating을 준 genre는 추천되지 않음을 알 수 있다. 다만 추천된 영화 중 3개의 영화는 comedy genre의 영화들인데 이는 한개의 영화가 여러 genre에 속하는 경우가 많아서 이러한 점이 반영된 것으로 생각할 수 있다. User 945는 가족 영화가 많이 추천된 것을 알 수 있다.

4 자신이 실습 결과 분석을 더 잘하기 위해 했던 방법들을 작성한다.

결과를 한눈에 잘 보기 위해 spark dataframe을 pandas dataframe으로 변환하는 등 pandas dataframe을 사용했다. 또한 위의 2번에서 코드 첨부한 내용과 동일하게 u.item 파일을 읽어들인 후 아래 코드를 사용하면 특정 장르의 영화를 8개 표시되게 할 수 있다.

```
import pandas as pd

# u.item 파일 경로를 지정
file_path = 'ml-100k/u.item'

# 데이터를 불러오기
column_names = ['movie_id', 'movie_title', 'release_date', 'video_release_date', 'IMDb_URL',
'unknown', 'Action', 'Adventure', 'Animation', 'Children\'s', 'Comedy', 'Crime', 'Documentary',
'Drama', 'Fantasy', 'Film-Noir', 'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi',
'Thriller', 'War', 'Western']
df = pd.read_csv(file_path, sep='|', names=column_names, encoding='latin-1')
# 필요한 열만 선택
selected_columns = df[['movie_id', 'movie_title', 'release_date']]


# 장르가 액션인 영화만 필터링
action_movies = selected_columns[df['Horror'] == 1]
# 액션 영화 데이터프레임 출력
print(action_movies.head(8))  # 첫 8개 행 확인

animation_movies = selected_columns[df['Thriller'] == 1]
print(animation_movies.head(8))  # 첫 8개 행 확인
```

```
     movie_id                                movie_title release_date
16         17                  From Dusk Till Dawn (1996)  05-Feb-1996
83         84  Robert A. Heinlein's The Puppet Masters (1994)  01-Jan-1994
100       101                           Heavy Metal (1981)  08-Mar-1981
122       123                      Frighteners, The (1996)  19-Jul-1996
182       183                                Alien (1979)  01-Jan-1979
183       184                      Army of Darkness (1993)  01-Jan-1993
184       185                                Psycho (1960)  01-Jan-1960
199       200                           Shining, The (1980)  01-Jan-1980
    movie_id                movie_title release_date
1          2             GoldenEye (1995)  01-Jan-1995
2          3             Four Rooms (1995)  01-Jan-1995
4          5               Copycat (1995)  01-Jan-1995
10        11         Seven (Se7en) (1995)  01-Jan-1995
11        12    Usual Suspects, The (1995)  14-Aug-1995
16        17    From Dusk Till Dawn (1996)  05-Feb-1996
20        21  Muppet Treasure Island (1996)  16-Feb-1996
22        23           Taxi Driver (1976)  16-Feb-1996
```

또한 다른 rating을 train에 넣고 추천 결과를 도출해 보았다. 그 결과 User 944는 1개의 영화를 제외하고는 Horror, Thriller, Drama, Mystery genre의 영화가 추천된다. User 945는 2개의 영화를 제외하고는 Comedy, Children's, Musical, Drama genre의 영화가 추천된다. u.item 파일을 dataframe으로 바꿔서 보면 영화 1개 당 여러 개의 genre에 해당하는 영화가 많다. 그래서 높은 평점을 준 영화 장르만 나오지 않고 비슷한 분위기의 장르의 영화들이 결과로 나온다. 또한 같은 Drama genre라 해도 FairyTale: A True Story같은 fantasy 영화도 있고 Postman, The 같은 apocalypse영화도 있다. 이런 점들을 감안한다면 결과가 대체로 잘 도출되었다고 생각할 수 있다.