Content Recommendation System Report

By:

Pratham Shah

Adam Afridi

Adil Shaikh

Shruti Shukla

1. Introduction

This report describes the development of a movie recommender system using the Amazon Web Services (AWS) platform. The goal is to create a system that suggests relevant movies to users based on their preferences and historical interactions.

2. Methodology

2.1. Technology Stack:

- **Amazon SageMaker:** Amazon SageMaker is a cloud-based machine-learning platform that allows the creation, training, and deployment by developers of machine-learning (ML) models on the cloud. It can be used to deploy ML models on embedded systems and edge-devices.
- **Amazon Personalize:** Amazon Personalize is a fully managed machine learning service that uses your data to generate item recommendations for your users. It can also generate user segments based on the users' affinity for certain items or item metadata.

2.2. Features / Recommenders made:

- **More like X Recommender:** Get recommendations for videos that are similar to a video you specify.
- **Top Picks for you Recommender:** Get personalized content recommendations for a user you specify.

3. Output:

In [46]:
```
# First pick a user
test_user_id = "2"

# Select a random item
test_item_id = "81847" #Iron Man 59315, Tangled: 81847

# Get recommendations for the user for this item
get_recommendations_response = personalize_runtime.get_recommendations(
    recommenderArn = recommender_more_like_x_arn,
    userId = test_user_id,
    itemId = test_item_id,
    numResults = 5
)

# Build a new dataframe for the recommendations
item_list = get_recommendations_response['itemList']
recommendation_list = []
for item in item_list:
    movie = get_movie_by_id(item['itemId'], items_df)
    recommendation_list.append(movie)

user_recommendations_df = pd.DataFrame(recommendation_list, columns = [get_movie_by_id(test_item_id, items_df)])

pd.options.display.max_rows = 20
display(user_recommendations_df)
```

	Tangled (2010)
0	Finding Nemo (2003)
1	Shrek (2001)
2	Toy Story 3 (2010)
3	How to Train Your Dragon (2010)
4	Hercules (1997)

In [48]:
```
# First pick a user
test_user_id = "111" # samples users: 55, 75, 76, 111

# Get recommendations for the user
get_recommendations_response = personalize_runtime.get_recommendations(
    recommenderArn = recommender_top_picks_arn,
    userId = test_user_id,
    numResults = 20
)

# Build a new dataframe for the recommendations
item_list = get_recommendations_response['itemList']
recommendation_list = []
for item in item_list:
    movie = get_movie_by_id(item['itemId'], items_df)
    recommendation_list.append(movie)

column_name = test_user_id+" ("+get_gender_by_id(test_user_id, users_data_df)+")"

user_recommendations_df = pd.DataFrame(recommendation_list, columns = [column_name])

pd.options.display.max_rows =20
display(user_recommendations_df)
```

	111 (male)
0	Cave of Forgotten Dreams (2010)
1	Social Network, The (2010)
2	Inside Job (2010)
3	Better Living Through Circuitry (1999)
4	Best Worst Movie (2009)
5	Spotlight (2015)
6	Henry's Crime (2010)
7	Inception (2010)
8	Somewhere (2010)
9	Steam of Life (Miesten vuoro) (2010)
10	Greenberg (2010)
11	Little Fockers (2010)
12	A-Team, The (2010)
13	Hellsinki (Rööperi) (2009)

4. Algorithms

- **SIMS (Similar Items):** This algorithm recommends movies similar to a given movie based on user interactions. It's useful for suggesting movies that have similar characteristics or appeal to users who have interacted with similar movies in the past.

- **User-based collaborative filtering:** Recommends movies to a user based on the preferences of similar users.

- **User-based collaborative filtering:** Recommends movies to a user based on the preferences of similar users.

5. Datasets:

```
USER_ID,ITEM_ID,TIMESTAMP,EVENT_TYPE
1,1,964982703,watch
1,3,964981247,watch
1,6,964982224,watch
1,47,964983815,watch
1,50,964982931,watch
1,101,964980868,watch
1,110,964982176,watch
1,151,964984041,watch
1,157,964984100,watch
1,163,964983650,watch
1,216,964981208,watch
1,231,964981179,watch
1,235,964980908,watch
1,260,964981680,watch
1,333,964981179,watch
1,349,964982563,watch
1,356,964980962,watch
1,362,964982588,watch
1,367,964981710,watch
1,441,964980868,watch
1,457,964981909,watch
1,480,964982346,watch
1,527,964984002,watch
1,543,964981179,watch
1,552,964982653,watch
1,553,964984153,watch
1,590,964982546,watch
1,592,964982271,watch
```

```
ITEM_ID,GENRES,YEAR,CREATION_TIMESTAMP
1,Adventure|Animation|Children|Comedy|Fantasy,1995,1640995200
2,Adventure|Children|Fantasy,1995,1640995200
3,Comedy|Romance,1995,1640995200
4,Comedy|Drama|Romance,1995,1640995200
5,Comedy,1995,1640995200
6,Action|Crime|Thriller,1995,1640995200
7,Comedy|Romance,1995,1640995200
8,Adventure|Children,1995,1640995200
9,Action,1995,1640995200
10,Action|Adventure|Thriller,1995,1640995200
11,Comedy|Drama|Romance,1995,1640995200
12,Comedy|Horror,1995,1640995200
13,Adventure|Animation|Children,1995,1640995200
14,Drama,1995,1640995200
15,Action|Adventure|Romance,1995,1640995200
16,Crime|Drama,1995,1640995200
17,Drama|Romance,1995,1640995200
18,Comedy,1995,1640995200
19,Comedy,1995,1640995200
20,Action|Comedy|Crime|Drama|Thriller,1995,1640995200
21,Comedy|Crime|Thriller,1995,1640995200
22,Crime|Drama|Horror|Mystery|Thriller,1995,1640995200
23,Action|Crime|Thriller,1995,1640995200
24,Drama|Sci-Fi,1995,1640995200
25,Drama|Romance,1995,1640995200
26,Drama,1995,1640995200
27,Children|Drama,1995,1640995200
28,Drama|Romance,1995,1640995200
29,Adventure|Drama|Fantasy|Mystery|Sci-Fi,1995,1640995200
30,Crime|Drama,1995,1640995200
31,Drama,1995,1640995200
32,Mystery|Sci-Fi|Thriller,1995,1640995200
34,Children|Drama,1995,1640995200
36.Crime|Drama.1995.1640995200
```

```
USER_ID,GENDER
1,male
2,female
3,male
4,male
5,male
6,male
7,male
8,male
9,male
10,male
11,male
12,male
13,female
14,male
15,male
16,male
17,female
18,male
19,male
20,male
21,female
22,female
23,female
24,female
25,male
26,male
27,male
28,male
29,female
30,male
31,male
32,male
33,female
34,female
```