Data Exploration

For this problem set, I studied a variety of movies and how users on a social media platform felt about those movies. I then developed five different recommendation methods in prolog, utilizing data from the movies dataset to recommend movies to a user. The data contained 67 unique movies, their director(s), the genre(s) (action, adventure, animation, biography, comedy, crime, drama, family, fantasy, music, romance, scifi, thriller, western), and the actors/actresses that starred in it. These 67 movies starred 1434 unique actors, and 62 unique directors.

The simulated social media platform contained 10 users, each with a set of different liked movies and a list of users they are friends with. Each user had about 4-6 friends. Of the 67 total movies, 38 movies were liked by the users with the most liked being Frozen and The Big Short, each liked by 3 out of 10 users which leaves 29 movies without any users who have liked them.

Implementation Justification

Each recommendation uses a different aspect of the dataset to recommend users similar movies to ones they have liked. As a general note, each recommendation includes a clause to ensure that users are not recommended movies they have already seen.

Recommendation 1 is a friend based recommendation. If one of the user's friends likes a movie, this predicate will recommend it to them. For instance, since User 1 and User 2 are friends, User 1 may like other movies User 2 has seen.

Recommendation 2 is a genre based recommendation. This predicate will recommend users movies in the same genre as the users liked movies. This predicate is straight forward as some users gravitate towards a certain genre, for instance, User 5 has liked all 3 movies in The Hobbit trilogy. Because of this, they may like other Fantasy films so this predicate will recommend fantasy films to them.

Recommendation 3 is a director based recommendation. This predicate will recommend any movies to users that have been directed by the same directors as their other liked movies. When people like films, they may find themselves gravitating towards a certain director, which is what this predicate addresses. For example, user 4 likes both *Interstellar* and *Inception*, so this predicate would recommend them other movies directed by Chrstopher Nolan like *The Dark Knight Rises*.

In a similar way, users may also gravitate towards movies that star a specific actor/actress, for this purpose, recommendation 4 is an actor/actress based recommendation that works very similarly to recommendation 3, but with actors/actresses instead of directors. This predicate will recommend any movies to users that have the same actors/actresses as their other liked movies. For instance, all of User 4's liked movies star Christian Bale. From that, it can be assumed that this user is a fan of Christian Bale and will be recommended other movies he has starred in.

Finally, recommendation 5 is a "similar user recommendation." If User X and User Y both liked a movie, it will recommend User X other movies User Y has liked and vice versa, even if User X and User Y are not friends. This implementation functions similarly to a "Users like you also liked…" algorithm that other streaming services use. This recommendation might be useful if a user doesn't have many friends on the social media platform.

Implementation Demonstrations

User 1:

Recommendation 1

```
?- recommendation1(user01, Movie).
Movie = footloose ;
Movie = diary_of_a_wimpy_kid ;
Movie = american_hustle ;
Movie = the_big_short ;
Movie = interstellar
```

(...)

These recommendations make sense as they are films User01's friends have liked.

Recommendation 2

```
?- recommendation2(user01, Movie).
Movie = diary_of_a_wimpy_kid ;
Movie = footloose ;
Movie = guardians_of_the_galaxy ;
Movie = hot_tub_time_machine ;
Movie = kick_ass
```

(...)

These recommendations make sense as these films are in the same genre as User 01's other liked films (in this case, Comedy).

Recommendation 3

```
?- recommendation3(user01,Movie).
false.
```

This result makes sense, there is only one movie directed by each director of User01's liked movies in the dataset OR this User has seen all of the directors movies in the dataset. This similar scenario occurs multiple times with this recommendation predicate.

Recommendation 4

```
?- recommendation4(user01,Movie).
Movie = the_conjuring ;
Movie = shutter_island ;
Movie = drive ;
Movie = the_big_short ;
Movie = the_big_short
```

(…)

In a similar manner to recommendation 3, this result makes sense as these movies all star an actor or actress that starred in User01's liked movies.

Recommendation 5

```
?- recommendation5(user01, Movie).
Movie = black_swan ;
Movie = gone_girl ;
Movie = melancholia ;
Movie = sleeping_beauty ;
Movie = the_hunger_games
```

(...)

These recommendations make sense as all these movies are liked by another user who also liked at least one of the same movies as User01, even if they are not friends.

User 2:

Recommendation 1

```
?- recommendation1(user02, Movie).
Movie = crazy_stupid_love ;
Movie = inside_out ;
Movie = rock_of_ages ;
Movie = the_grand_budapest_hotel ;
Movie = divergent
```

(...)
These recommendations make sense as they are films User02's friends have liked.

Recommendation 2

```
?- recommendation2(user02, Movie).
Movie = crazy_stupid_love ;
Movie = guardians_of_the_galaxy ;
Movie = hot_tub_time_machine ;
Movie = inside_out ;
Movie = kick_ass
```

(...)
These recommendations make sense as these films are in the same genre as User 02's other liked films (in this case, Comedy)

Recommendation 3

```
?- recommendation3(user02,Movie).
false.
```

Recommendation 4

```
?- recommendation4(user02,Movie).
Movie = rock_of_ages ;
Movie = gone_girl ;
Movie = divergent ;
Movie = fantastic_four ;
Movie = scott_pilgrim_vs_the_world
```

This result makes sense as these movies all star an actor or actress that starred in User02's liked movies.

(...)

Recommendation 5

```
?- recommendation5(user02, Movie).
Movie = crazy_stupid_love ;
Movie = inside_out ;
Movie = rock_of_ages ;
Movie = the_grand_budapest_hotel ;
Movie = crazy_stupid_love
```

(...)

These recommendations make sense as all these movies are liked by another user who also liked at least one of the same movies as User02, even if they are not friends.

User 3:

Recommendation 1

```
?- recommendation1(user03, Movie).
Movie = footloose ;
Movie = pitch_perfect ;
Movie = diary_of_a_wimpy_kid ;
Movie = frozen ;
Movie = american_hustle
```

(...)

These recommendations make sense as they are films User03's friends have liked.

Recommendation 2

```
?- recommendation2(user03, Movie).
Movie = avengers_age_of_ultron ;
Movie = captain_america_the_first_avenger
Movie = drive ;
Movie = fantastic_four ;
Movie = guardians_of_the_galaxy
```

(...)

These recommendations make sense as these films are in the same genre as User 03's other liked films (in this case, Action).

Recommendation 3

```
?- recommendation3(user03,Movie).
Movie = the_social_network ;
Movie = the_martian ;
false.
```

These recommendations make sense as they are directed by other directors of User 03's liked films, in this case, the options are exhausted after 2 results.

Recommendation 4

```
?- recommendation4(user03,Movie).
Movie = fantastic_four ;
Movie = footloose ;
Movie = thor ;
Movie = mad_max_fury_road ;
Movie = footloose
```

(...)

This result makes sense as these movies all star an actor or actress that starred in User03's liked movies.

Recommendation 5

```
?- recommendation5(user03, Movie).
Movie = black_swan ;
Movie = crazy_stupid_love ;
Movie = frozen ;
Movie = melancholia ;
Movie = the_hunger_games
```

(...)

These recommendations make sense as all these movies are liked by another user who also liked at least one of the same movies as User03, even if they are not friends.

User 4:

Recommendation 1

```
?- recommendation1(user04, Movie).
Movie = crazy_stupid_love ;
Movie = frozen ;
Movie = inside_out ;
Movie = pitch_perfect ;
Movie = rock_of_ages
```

(...)
These recommendations make sense as they are films User04's friends have liked .

Recommendation 2

```
?- recommendation2(user04, Movie).
Movie = john_wick ;
Movie = kick_ass ;
Movie = killing_them_softly ;
Movie = legend ;
Movie = nightcrawler
```

(...)

These recommendations make sense as these films are in the same genre as User 04's other liked films (In this case, crime).

Recommendation 3

```
?- recommendation3(user04,Movie).
Movie = the_dark_knight_rises ;
Movie = the_dark_knight_rises.
```

(...)

This makes sense, as this user has liked 2 Chirstopher Nolan movies, so *The Dark Knight Rises* appears twice, and that is the other Nolan movie in the dataset.

Recommendation 4

```
?- recommendation4(user04, Movie).
Movie = kick_ass ;
Movie = her ;
Movie = guardians_of_the_galaxy ;
Movie = the_dark_knight_rises ;
Movie = knock_knock
```

(...)

This result makes sense as these movies all star an actor or actress that starred in User04's liked movies.

Recommendation 5

```
?- recommendation5(user04, Movie).
Movie = legend ;
Movie = the_imitation_game ;
Movie = the_social_network ;
Movie = the_wolf_of_wall_street ;
Movie = ex_machina
```

(...)

These recommendations make sense as all these movies are liked by another user who also liked at least one of the same movies as User04, even if they are not friends.

User 5:

Recommendation 1

```
?- recommendation1(user05, Movie).
Movie = crazy_stupid_love ;
Movie = frozen ;
Movie = inside_out ;
Movie = pitch_perfect ;
Movie = rock_of_ages
```

(...)

These recommendations make sense as they are films User05's friends have liked.

Recommendation 2

```
?- recommendation2(user05, Movie).
Movie = avengers_age_of_ultron ;
Movie = captain_america_the_first_avenger ;
Movie = divergent ;
Movie = fantastic_four ;
Movie = frozen
```

(...)

These recommendations make sense as these films are in the same genre as User 05's other liked films (In this case, fantasy).

Recommendation 3

```
?- recommendation3(user05,Movie).
false.
```

This result makes sense, there is only one movie directed by each director of User03's liked movies in the dataset OR this User has seen all of the directors movies in the dataset.

Recommendation 4

```
?- recommendation4(user05, Movie).
Movie = avengers_age_of_ultron ;
Movie = the_imitation_game ;
Movie = captain_america_the_first_avenger ;
Movie = x_men_days_of_future_past ;
Movie = guardians_of_the_galaxy
```

(...)

This result makes sense as these movies all star an actor or actress that starred in User05's liked movies.

Recommendation 5

```
?- recommendation5(user05, Movie).
false.
```

This result makes sense as no other users have listed *The Hobbit Trilogy as a liked film.*

User 6:

Recommendation 1

```
?- recommendation1(user06, Movie).
Movie = crazy_stupid_love ;
Movie = frozen ;
Movie = inside_out ;
Movie = pitch_perfect ;
Movie = rock_of_ages
```

(...)

These recommendations make sense as they are films User06's friends have liked.

Recommendation 2

```
?- recommendation2(user06, Movie).
Movie = captain_america_the_first_avenger ;
Movie = divergent ;
Movie = drive ;
Movie = inception ;
Movie = jack_reacher
```

(...)

These recommendations make sense as these films are in the same genre as User 06's other liked films (In this case, action).

Recommendation 3

```
?- recommendation3(user06,Movie).
false.
```

This result makes sense, there is only one movie directed by each director of User06's liked movies in the dataset OR this User has seen all of the directors movies in the dataset.

Recommendation 4

```
?- recommendation4(user06, Movie).
Movie = the_hobbit_an_unexpected_journey ;
Movie = captain_america_the_first_avenger ;
Movie = the_cabin_in_the_woods ;
Movie = thor ;
Movie = captain_america_the_first_avenger
```

(...)

This result makes sense as these movies all star an actor or actress that starred in User06's liked movies.

Recommendation 5

```
?- recommendation5(user06, Movie).
false.
```

No other users liked the same movies as User06.

User 7:

Recommendation 1

```
?- recommendation1(user07, Movie).
Movie = footloose ;
Movie = pitch_perfect ;
Movie = diary_of_a_wimpy_kid ;
Movie = divergent ;
Movie = jack_reacher
```

(...)

These recommendations make sense as they are films User07's friends have liked.

Recommendation 2

```
?- recommendation2(user07, Movie).
Movie = american_hustle ;
Movie = diary_of_a_wimpy_kid ;
Movie = django_unchained ;
Movie = drive ;
Movie = ex_machina
```

(...)

Recommendation 3

```
?- recommendation3(user07,Movie).
Movie = the_social_network ;
false.
```

This result makes sense, there is only one movie directed by each director of User07's liked movies in the dataset OR this User has seen all of the directors movies in the dataset.

Recommendation 4

```
?- recommendation4(user07, Movie).
Movie = thor ;
Movie = captain_america_the_first_avenger ;
Movie = hot_tub_time_machine ;
Movie = the_martian ;
Movie = the_conjuring
```

(...)

This result makes sense as these movies all star an actor or actress that starred in User07's liked movies.

Recommendation 5

```
?- recommendation5(user07, Movie).
Movie = inside_out ;
Movie = pitch_perfect ;
Movie = rock_of_ages ;
Movie = the_grand_budapest_hotel ;
Movie = inside_out
```

(...)

These recommendations make sense as all these movies are liked by another user who also liked at least one of the same movies as User07, even if they are not friends.

User 8:

Recommendation 1

```
?- recommendation1(user08, Movie).
Movie = footloose ;
Movie = pitch_perfect ;
Movie = diary_of_a_wimpy_kid ;
Movie = frozen ;
Movie = divergent
```

(...)

These recommendations make sense as they are films User08's friends have liked.

Recommendation 2

```
?- recommendation2(user08, Movie).
Movie = the_intouchables ;
Movie = the_kings_speech ;
Movie = american_hustle ;
Movie = john_wick ;
Movie = kick_ass
```

(...)

Recommendation 3

```
?- recommendation3(user08,Movie).
Movie = gone_girl ;
Movie = shutter_island ;
false.
```

Similar case to User 4, this predicate displays all movies directed by the directors of User08's liked movies they haven't already seen and then the options are exhausted.

Recommendation 4

```
?- recommendation4(user08, Movie).
Movie = harry_potter_and_the_deathly_hallows_part_1 ;
Movie = harry_potter_and_the_deathly_hallows_part_2 ;
Movie = sleeping_beauty ;
Movie = the_revenant ;
Movie = captain_america_the_first_avenger
```

(...)

This result makes sense as these movies all star an actor or actress that starred in User08's liked movies.

Recommendation 5

```
?- recommendation5(user08, Movie).
Movie = american_hustle ;
Movie = interstellar ;
Movie = inception ;
Movie = ex_machina ;
Movie = her
```

(...)

These recommendations make sense as all these movies are liked by another user who also liked at least one of the same movies as User08, even if they are not friends.

User 9:

Recommendation 1

```
?- recommendation1(user09, Movie).
Movie = crazy_stupid_love ;
Movie = frozen ;
Movie = inside_out ;
Movie = pitch_perfect ;
Movie = rock_of_ages
```

(...)

These recommendations make sense as they are films User09's friends have liked.

Recommendation 2

```
?- recommendation2(user09, Movie).
Movie = american_hustle ;
Movie = black_swan ;
Movie = crazy_stupid_love ;
Movie = diary_of_a_wimpy_kid ;
Movie = django_unchained
```

(...)

Recommendation 3

```
?- recommendation3(user09,Movie).
Movie = shutter_island ;
false.
```

Similar to User 4 & 8, this predicate displays all movies directed by the directors of User09's liked movies they haven't already seen and then the options are exhausted.

Recommendation 4

```
?- recommendation4(user09, Movie).
Movie = kick_ass ;
Movie = harry_potter_and_the_deathly_hallows_part_1 ;
Movie = harry_potter_and_the_deathly_hallows_part_2 ;
Movie = the_revenant ;
Movie = drive
```

(...)

This result makes sense as these movies all star an actor or actress that starred in User09's liked movies.

Recommendation 5

```
?- recommendation5(user09, Movie).
Movie = american_hustle ;
Movie = interstellar ;
Movie = inception ;
Movie = legend ;
Movie = the_social_network
```

(...)

These recommendations make sense as all these movies are liked by another user who also liked at least one of the same movies as User09, even if they are not friends.

User 10:

Recommendation 1

```
?- recommendation1(user10, Movie).
Movie = crazy_stupid_love ;
Movie = frozen ;
Movie = inside_out ;
Movie = pitch_perfect ;
Movie = rock_of_ages
```

(...)

These recommendations make sense as they are films User10's friends have liked.

Recommendation 2

```
?- recommendation2(user10, Movie).
Movie = american_hustle ;
Movie = black_swan ;
Movie = crazy_stupid_love ;
Movie = diary_of_a_wimpy_kid ;
Movie = django_unchained
```

(...)

Recommendation 3

```
?- recommendation3(user10,Movie).
false.
```

This result makes sense, there is only one movie directed by each director of User10's liked movies in the dataset OR this User has seen all of the directors movies in the dataset.

Recommendation 4

```
?- recommendation4(user10, Movie).
Movie = avengers_age_of_ultron ;
Movie = the_avengers ;
Movie = captain_america_the_first_avenger ;
Movie = the_hunger_games ;
Movie = jurassic_world ;
Movie = thor
```

This result makes sense as these movies all star an actor or actress that starred in User04's liked movies.

(...)

Recommendation 5

```
?- recommendation5(user10, Movie).
false.
```

No other users have liked the same movies as User 10.