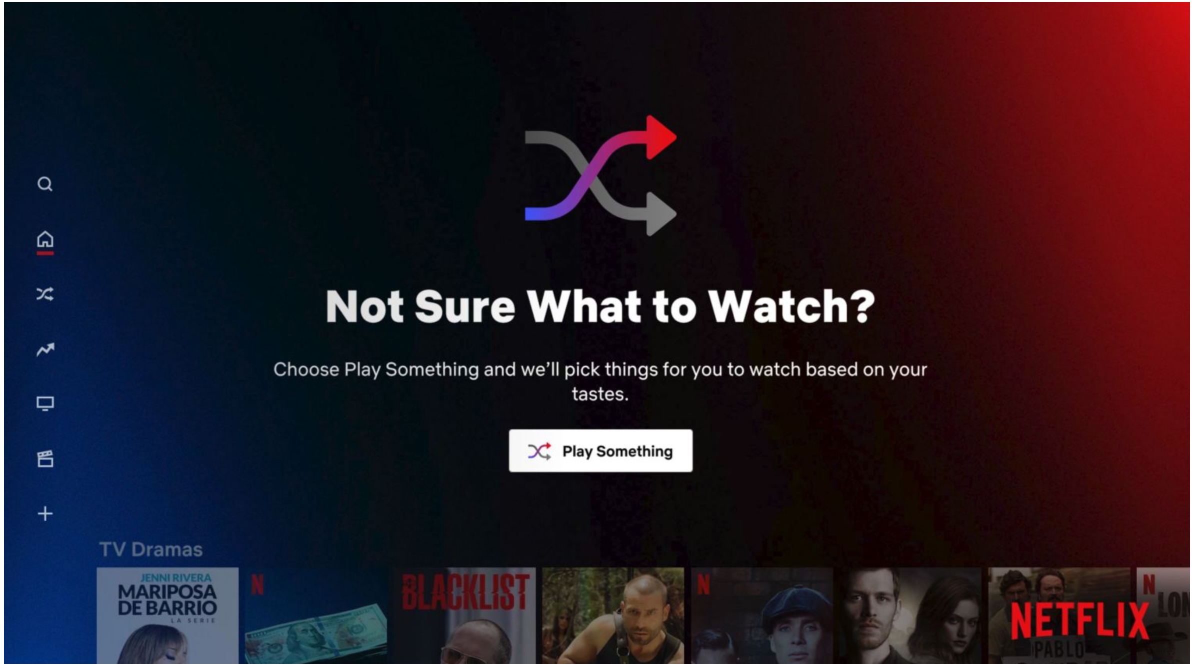
Entertainment 6: Next on Netflix

Kyle Brown, Jarrett Fein, Keara Hayes,
Jana Ratzloff, Sania Sinha, and Dan Smieszny

CMSE 202 Section 3
4/26/2022

Project Background

1st world problem: you don't know what to watch next on Netflix



Project Premises

Original Task:

- *Predict* the *genre* of the next TV show or movie of a Netflix user.

Modified Task:

- *Recommend* the next movie and/or show to watch
- *Recommend* multiple based on entire watch history

Considerations:

- Watch next is based off the one previous thing you just watched
- Netflix Home is based off your entire watch history
- Watch history is all of the previous things you've watched

Our Approach

Make several different recommendation systems and compare the results

- Jana's system is based on movie ratings
- Keara's system uses a movie or show title, description, cast and director
- Sania's system uses movie director, plot summary and genre
- Jarrett's system uses data given in a user's "NetflixViewingHistory.csv" file
- Dan's system uses title similarity and genre

Packages used: sklearn, pandas, numpy, matplotlib, ast, difflib

Netflix watch history data courtesy of: Dan, Jarrett, Kyle

Jana’s System - Recommendations Based on Ratings

- Movies dataset gives each movie title an ID
- Ratings dataset shows each users rating for each movie identified by ID
  - Ratings are 0-5, 5 = best, 0 = user did not rate that movie
- Combine into one big dataset
- Not all ratings are equal
  - Ex: restaurant with one hundred 5-star reviews >> restaurant with one 5-star reviews

movieId	title
1	Toy Story (1995)
2	Jumanji (1995)
3	Grumpier Old Men (1995)
4	Waiting to Exhale (1995)
5	Father of the Bride Part II (1995)

userId	movieId	rating
1	1	4.0
1	3	4.0
1	6	4.0
1	47	5.0
1	50	5.0

userId	1	2	3	4	5	6	7	8	9	10
movieId										
1	4.0	0.0	0.0	0.0	4.0	0.0	4.5	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	4.0	0.0	4.0	0.0	0.0
3	4.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0

https://grouplens.org/datasets/movielens/latest/

Jana’s System - Recommendations Based on Ratings

```python
def function(user_threshold, movie_threshold, movie_name, number_to_recommend):
```

- user_threshold = # of people that voted for a movie to qualify that movie
- movie_threshold = # of movies a user needs to have voted for to qualify their opinions
- movie_name = manual input of the movie title
  - Recommendation system is limited to movies within the dataset
- number_to_recommend = # of movies we want the recommendation system to output
- System uses the KNN algorithm to compute similarity with cosine distance metric
  - The closer the distance is to 1, the higher up on the recommendation list

Jana's System - Recommendations Based on Ratings

Star Trek and Stuart Little were imputed into the system with different user thresholds (# of people that voted for a movie to qualify that movie)

```
1 function(20, 50, 'Star Trek', 10,
```

	Title	Distance
1	Star Trek IV: The Voyage Home (1986)	0.472060
2	Jurassic Park (1993)	0.471171
3	Batman (1989)	0.457484
4	Batman Forever (1995)	0.455311
5	True Lies (1994)	0.449626
6	Fugitive, The (1993)	0.448126
7	Mask, The (1994)	0.447216
8	Clear and Present Danger (1994)	0.445350
9	Star Trek: First Contact (1996)	0.421643
10	Stargate (1994)	0.390724

```
1 function(50, 50, 'Star Trek', 10,
```

	Title	Distance
1	Cliffhanger (1993)	0.473866
2	Jurassic Park (1993)	0.471171
3	Batman (1989)	0.457484
4	Batman Forever (1995)	0.455311
5	True Lies (1994)	0.449626
6	Fugitive, The (1993)	0.448126
7	Mask, The (1994)	0.447216
8	Clear and Present Danger (1994)	0.445350
9	Star Trek: First Contact (1996)	0.421643
10	Stargate (1994)	0.390724

```
1 function(10, 50, 'Stuart Little', 10, plot
```

	Title	Distance
1	Muppets Take Manhattan, The (1984)	0.624958
2	Bambi (1942)	0.617884
3	Family Man, The (2000)	0.607573
4	Sleeping Beauty (1959)	0.604447
5	Holes (2003)	0.600212
6	Lady and the Tramp (1955)	0.591959
7	Dinosaur (2000)	0.578952
8	101 Dalmatians (One Hundred and One Dalmatians...	0.572271
9	My Dog Skip (1999)	0.526798
10	Kid, The (2000)	0.486673

```
1 function(15, 50, 'Stuart Little', 10, plot
```

	Title	Distance
1	Angels in the Outfield (1994)	0.645794
2	Bring It On (2000)	0.643263
3	Perfect Storm, The (2000)	0.642960
4	Bug's Life, A (1998)	0.635016
5	Bambi (1942)	0.617884
6	Family Man, The (2000)	0.607573
7	Sleeping Beauty (1959)	0.604447
8	Holes (2003)	0.600212
9	Lady and the Tramp (1955)	0.591959
10	101 Dalmatians (One Hundred and One Dalmatians...	0.572271

Keara’s Watch Next - Recommendations Using TFIDF

TFIDF - Term Frequency Inverse Document Frequency (or, word frequency, in short)

- You feed the function a film or show, and it spits out a recommendation.
- Recommendation based on similarities between title, description, cast, and director
- Gives films and shows similarity scores for each metric, and these scores are combined to recommend the “best” option.

Keara's Watch Next - Recommendations Using TFIDF

It works pretty well, too!

```
rec = movie_rec('Stuart Little')
print(rec)

# Stuart Little 2 is the sequel to Stuart Little, so this is reasonable
```

```
Stuart Little 2
```

```
rec = movie_rec('Ash vs. Evil Dead')
print(rec)

# 'Ash vs. Evil Dead' is a spinoff of the 'The Evil Dead' movie franchise, so this is
# definitely a good recommendation.
```

```
The Evil Dead
```

Keara's Watch Next - Recommendations Using TFIDF

And, since there's so much Star Trek in Jarrett's watch history:

```
rec = movie_rec('Star Trek: The Next Generation')
print(rec)

# While Deep Space Nine is probably a better recommendation, getting in the right franchise
# is a good sign, and it picked up on which starship to focus on
```

```
Star Trek: Enterprise
```

```
rec = movie_rec('Star Trek')
print(rec)

# 'For the Love of Spock' is a documentary about Star Trek and Leonard Nimoy, so this is
# a reasonable recommendation if you've just finished the JJ Abrams Star Trek reboot movie
```

```
For the Love of Spock
```

Keara’s Watch Next - Recommendations Using TFIDF

Limitations:

- There's a chance that a movie/show won't have a perfect correlation with itself, so it isn't excluded from the recommendation list and may accidentally be recommended to the user.
- Sometimes the descriptions in the csv used for this aren't very good, leading to erroneous recommendations.

Jarrett's Netflix Home: Netflix User Data

- All Netflix users have access to their watch history via a file called "NetflixViewingHistory.csv"
- Only recorded data that Netflix provides to users; they likely collect more

Title	Date
Marvel's Daredevil: Season 3: Revelations	2/28/2022
Marvel's Daredevil: Season 3: Upstairs/Downstairs	2/28/2022
Marvel's Daredevil: Season 3: Aftermath	2/24/2022
Marvel's Daredevil: Season 3: The Devil You Know	2/24/2022
Marvel's Daredevil: Season 3: The Perfect Game	2/21/2022
Marvel's Daredevil: Season 3: Blindsided	2/16/2022
Marvel's Daredevil: Season 3: No Good Deed	2/14/2022
Marvel's Daredevil: Season 3: Please	2/11/2022
Marvel's Daredevil: Season 3: Resurrection	2/9/2022

Star Trek: The Next Generation: Season 4: Final Mission	7/29/2021
Star Trek: The Next Generation: Season 4: Future Imperfect	7/28/2021
Star Trek: The Next Generation: Season 4: Reunion	7/28/2021
Star Trek: The Next Generation: Season 4: Legacy	7/26/2021
Star Trek: The Next Generation: Season 4: Remember Me	7/25/2021
Star Trek: The Next Generation: Season 4: Suddenly Human	7/23/2021
Star Trek: The Next Generation: Season 4: Brothers	7/23/2021
For the Love of Spock	7/22/2021
Star Trek: The Next Generation: Season 4: Family	7/22/2021
Star Trek: The Next Generation: Season 4: The Best of Both Worlds: Part 2	7/19/2021

The Queen's Gambit: Limited Series: Openings	10/29/2020
The Social Dilemma	10/26/2020
The West Wing: Season 1: Pilot	10/26/2020
The Universe: Season 2: Alien Planets	10/26/2020
Jeopardy!: Seth Wilson Collection: Episode #7361	4/22/2020
Tiger King: Not Your Average Joe	4/22/2020

Not a lot of information to go on!

Jarrett's Netflix Home: Netflix User Data

- Take note of words in titles
- Count number of episodes of a given series
- Calculate number of days since since watching
- Remove redundant information

```
([["marvel's", 'daredevil'], ["marvel's", 'defenders'], ['star', 'trek'], ['love', 'spock'], ['get', 'roger', 'stone'], ["quee
n's", 'gambit'], ['social', 'dilemma'], ['west', 'wing'], ['universe'], ['jeopardy!'], ['tiger', 'king']], [35, 8, 108, 1, 1,
7, 1, 1, 1, 1, 1], [93.51428571428572, 79.625, 232.94444444444446, 277.0, 398.0, 540.4285714285714, 546.0, 546.0, 546.0, 733.0,
733.0])
```

From this, we can comb the Netflix library for movies and shows with similar titles (i.e. contains the same words) and assign them weights based on the numbers we've extracted.

Weights will be "episodes per day" of each show in our history. Shows and movies similar in title to highly weighted watched items in our history will be rated higher.

Jarrett's Netflix Home: Netflix User Data

Results:

```
[['bright', 'star'],
 ['star', 'trek'],
 ['holly', 'star'],
 ['5', 'star', 'christmas'],
 ['super', 'monsters', 'wish', 'star'],
 ['look', 'star'],
 ['puppy', 'star', 'christmas'],
 ['pup', 'star'],
 ['star', 'trek'],
 ['pup', 'star'],
 ['star', 'trek'],
 ['star', 'trek'],
 ['frat', 'star'],
 ['star', 'trek'],
 ['pup', 'star'],
 ['barbie', 'star', 'light', 'adventure'],
 ['star', 'men'],
 ['star', 'wars'],
 ["marvel's", 'agents', 's.h.i.e.l.d.'],
 ["marvel's", 'jessica', 'jones'],
 ["marvel's", 'punisher'],
 ["marvel's", 'daredevil'],
 ["marvel's", 'iron', 'fist'],
 ["marvel's", 'luke', 'cage'],
 ["marvel's", 'defenders'],
 ["marvel's", 'hulk'],
 ["marvel's", 'iron', 'man', '&', 'hulk'],
 ['creating', "queen's", 'gambit'],
 ["queen's", 'gambit'],
 ['love', 'spectrum'],
 ['love', 'cost', 'thing'],
 ['love', 'puff'],
 ['really', 'love'],
```

- Some titles are repeated, this is expected and not an issue
- Some recommendations appear illogical based on watch history, but make sense in context of the algorithm
- Each show in my history has a weight assigned to it, not the associated recommendations
- This is not the complete list of recommendations, just the highest ranked

Sanya's Netflix Home: Cosine Similarity Matrix

In [9]:

```
# Selecting the features we want to use
features = ["director", "list", "description"]

for feature in features:
    movies_df[feature] = movies_df[feature].apply(literal_return)

movies_df[features].head(10)
```

Out[9]:

	director	list	description
0	Paul Verhoeven	Action & Adventure, Sci-Fi & Fantasy	After getting a memory implant, working stiff ...
1	Jennifer Kaytin Robinson	Comedies, Romantic Movies	On the heels of a blindsiding breakup, music j...
2	J. Lee Thompson	Action & Adventure, Classic Movies	During World War II, British forces launch an ...
3	Ivan Reitman	Classic Movies, Comedies, Cult Movies	After losing everything, an indolent sad sack ...
4	Sergio Pablos	Children & Family Movies, Comedies	A selfish postman and a reclusive toymaker for...
5	Paul Verhoeven	Action & Adventure, Sci-Fi & Fantasy	After getting a memory implant, working stiff ...
6	Terry Jones	Classic Movies, Comedies, Cult Movies	Born in a stable in Judea, Brian grows up to j...
7	David Gordon Green	Action & Adventure, Comedies	After witnessing a murder, a perpetually stone...
8	Mike Rianda, Jeff Rowe	Children & Family Movies, Comedies	A robot apocalypse put the brakes on their cro...
9	Matt Thompson	Action & Adventure, Comedies	A chainsaw-wielding George Washington teams wi...

- Build a 'soup' of information from all previously watched shows
- Features used: **director, genre, plot summary keywords**
- Final database created of 'soups' of watched and unwatched movies

Sanya’s Netflix Home: Cosine Similarity Matrix

```
In [16]:
# Metadata creation for final big database
def create_soup(features):
    return ' '.join(features['director'].split(',')) + ' ' + ' '.join(features['listed_in'].split(',')) + ' ' + ' '.join(features['descrip


movie_database["soup"] = movie_database.apply(create_soup, axis=1)
print(movie_database["soup"].head())
```

```
0     kirstenjohnson documentaries asherfathernearst...
6     robertcullen joséluisucha children&familymovie...
7     hailegerima dramas independentmovies internati...
9     theodoremelfi comedies dramas awomanadjustingt...
12    christianschwochow dramas internationalmovies ...
Name: soup, dtype: object
```

- Similarity matrix constructed
- Select top 10 similar entries except top ranked one, for index representing previously watched movies
- Top ranked movie is identical to soup of all watched movies

Sanya’s Netflix Home: Final Recommendations

- Works pretty well
- Multiple movies starring Bruce Willis despite cast not being a feature

	title	director	cast	list	description
0	Total Recall	Paul Verhoeven	Arnold Schwarzenegger, Rachel Ticotin, Sharon ...	Action & Adventure, Sci-Fi & Fantasy	After getting a memory implant, working stiff ...
1	Someone Great	Jennifer Kaytin Robinson	Gina Rodriguez, Brittany Snow, DeWanda Wise, L...	Comedies, Romantic Movies	On the heels of a blindsiding breakup, music j...
2	The Guns of Navarone	J. Lee Thompson	Gregory Peck, David Niven, Anthony Quinn, Stan...	Action & Adventure, Classic Movies	During World War II, British forces launch an ...
3	Stripes	Ivan Reitman	Bill Murray, Harold Ramis, Warren Oates, P.J. ...	Classic Movies, Comedies, Cult Movies	After losing everything, an indolent sad sack ...
4	Klaus	Sergio Pablos	Jason Schwartzman, J.K. Simmons, Rashida Jones...	Children & Family Movies, Comedies	A selfish postman and a reclusive toymaker for...
...	...	...	...	...	...
162	The November Man	Roger Donaldson	Pierce Brosnan, Luke Bracey, Olga Kurylenko, E...	Action & Adventure	An ex-CIA agent emerges from retirement to pro...
163	Kevin Hart: Let Me Explain	Leslie Small, Tim Story	Kevin Hart	Stand-Up Comedy	Philadelphia funnyman Kevin Hart takes the sta...
164	Bill Burr: Let It Go	Shannon Hartman	Bill Burr	Stand-Up Comedy	The musings of comedian Bill Burr are let loos...
165	Blackfish	Gabriela Cowperthwaite	NaN	Documentaries	This fascinating documentary examines the life...
166	Red Dawn	John Milius	Patrick Swayze, C. Thomas Howell, Lea Thompson...	Action & Adventure, Cult Movies	A group of teenagers witnesses Soviet and Cuba...

167 rows × 5 columns

```
################ Content Based System ########
Recommendations for Netflix Home
2480                  Shortcut Safari
3769                      The Do-Over
5474                  Sherlock Holmes
228                The Last Boy Scout
514                  Starsky & Hutch
617              The Whole Nine Yards
3192                  Game Over, Man!
5830                   The Other Guys
6119     You Don't Mess with the Zohan
1246               Free State of Jones
Name: title, dtype: object
```

Dan’s Netflix Home: Title Similarity & Genre

- Give recommendations on how similar titles are to watch history, and their genre
- Using similarity, select titles similar to those in the watch history, from the netflix data
- Loop through all movies found and collect their genres
- Once you have the genres construct a dictionary of genre: [movies]
- For the top ten most genres find the titles that are most similar to watch history and display the top 20

```
bestMatches = dan['Title'].apply(lambda x: difflib.get_close_matches(x, netflix['title']))
bestMatches.head(10)
```

	Title
0	[Big Time Rush, Big Time, Big Kill]
1	[Monster Island, Monster House, Super M...
2	[]
3	[Brand New Cherry Flavor]
4	[Starship Troopers: Traitor of Mars, St...
5	[I Think You Should Leave with Tim Robi...
6	[]
7	[I Think You Should Leave with Tim Robi...
8	[]

```
genres = []
for i in range(len(bestMatches)): # loop through series
    itemGenres = []
    for j in range(len(bestMatches[i])): #loop through list of similar movies
        itemGenres.append(netflix['listed_in'].values[netflix['title'] == bestMatches[i][j]]) #finds movie and appends
    if len(itemGenres) > 0: #if similar movies and their genres were found
        for arr in itemGenres: #loops through list of arrays of genres
            genres += arr[0].replace(" ", "").split(",") #remove white space and splits string to list and adds to genres
```

Dan’s Netflix Home: Title Similarity & Genre

```
closeMovies = {}
for i in range(10): #loops through top 10 genres
    key = listOfGenres[i] #key is the genre
    movies = genresAndMovies.get(key) #gets the list of movies for the genre
    closeMovies[key] = [] #sets genre equal to empty list fo teh similar movies
    listOfLists = list(dan["Title"].apply(lambda x: difflib.get_close_matches(x, movies))) #list of lists of similar
    for subList in listOfLists: #loops through lists of lists and adds sublist items to the closeMovies list
        closeMovies[key] += subList
```

```
Action&Adventure
['The Dealer', 'The Prince', 'The Saint', 'The Prison', 'Indiana Jones and the Raiders of the Lost Ark', 'The Losers', 'The Brave', 'The Take', 'The Art of the Steal', 'Black Beach', 'The Art of War', 'The Crow', 'The Decline', 'The Tuxedo', 'Ninja Assassin', 'Takers', 'The Killer', 'Solo: A Star Wars Story', 'Casino Royale', 'American Assassin']

Dramas
['The Hater', 'The Water Man', 'The Show', 'The Rainmaker', 'The Death of Stalin', 'Soldier', 'The Lovers', 'The Founder', 'Fatima', 'The Art of Loving', 'The Stolen', 'The Dirt', 'The Giant', 'The Son', 'Black Rose', 'Sabotage', 'The Dancer', 'The Land of Cards', 'The Reconquest', 'Greater']

InternationalMovies
['The Rover', 'The Hater', 'The Tenth Man', 'Soldier', 'Black Beach', 'The Rezort', 'The Other', 'The Sunshine Makers', 'Bangistan', 'Watchman', 'The Son', 'The Pianist', 'The Promise', 'Milea', 'The Square', 'The Prison', 'Hostages', 'Reaction', 'The Giant', 'The Incident']

Comedies
['The Dirt', 'The Lovers', 'The Tribe', 'The Player', 'The Package', 'The Death of Stalin', 'The Cruise', 'Soldier', 'Blue Mountain State: The Rise of Thadland', 'The Art of the Steal', 'The Gathering', 'The Informant!', 'The Motive', 'The Trap', 'The Starling', 'Cake', 'Watchman', 'Swearnet: The Movie', 'The Four Seasons', 'Blue Streak']
```

Dan’s Netflix Home: Title Similarity & Genre

Limitations:

- Having stopwords in the titles (eg. “The”) can add bias when looking at title similarity
    - One fix would removing specific stop words from the title
- Not all titles in watch history were in the netflix database, which led to using similarity to find genres of items in watch history

Conclusions, Limitations & Future Scope

- Recommendation systems could make predictions
  - How successful were they? It’s up to the opinion of a particular Netflix user
- There is not dataset with all of the movies or TV shows on Netflix
  - Netflix is constantly adding and removing content
- Possibility of combining approach based on recency and duration of watch and cosine similarity approach for Netflix Home

Check us out on GitHub!

NETFLIX

THANKS FOR WATCHING