streamOn

TV Show and Streaming Platform Recommendation System

Riya Jaggi (rj356), Divya Damodaran (dd492), Siddhi Chordia (sc2538),
Sidharth Vadduri (sv352), Kendall Lane (kal255)

Prototype 1: https://stream-on-proto1.herokuapp.com
Prototype 2: https://stream-on-proto2.herokuapp.com
Final App: http://stream-on.herokuapp.com

TABLE OF CONTENTS

Project Prototype (P05): Final App

Improvements Statement

Write 2-3 paragraphs explaining the changes you have done since the first AND the second prototype. Organize your discussion by the 3 components IR/ML/Social.

Information Retrieval (IR):
Our information retrieval aspect has grown significantly since our last two prototypes. In our first prototype, the results were solely based on the similar tv show input for only a subset of tv shows retrieving information from their transcripts. In our second prototype, the app was retrieving information based on the similar tv show and ad hoc inputs from descriptions, reviews, and transcripts. For our final app, we have fine tuned the retrieval by have slider inputs to decide the weighting the description, reviews, transcripts, and ad hoc search, including filters like streaming platforms, years, seasons, and genre, and allowing the user to choose to input shows and keywords that they do not want in their results. Our final app also had enhanced search and better UI.

Machine Learning (ML):

We implemented relevance feedback for our machine learning component. When receiving results the user will indicate whether they found the document to be relevant or not. We will then store this information in a json file. If the percentage of users saying a result is relevant for a show is >=0.75 then the result is added as a result doc for the show, and if the percentage is less than =<0.25 then the result is added as an irrelevant doc for the show. We additionally added that at least four users must vote for a result to be sorted into either category.

We also implemented this relevance metric for free search in addition to TV show search.

We then use this dictionary of relevant and irrelevant docs for a show to power our rocchio update algorithm for Transcripts. And in addition to rocchio for the free search we implement a metric where if a show is relevant it is boosted by its relevance score and if it is irrelevant it is negatively boosted by its irrelevance search.

Social Aspect:
We started implementing our Social Aspect in our second prototype, when we started using reviews to affect our information retrieval. We had the review similarity done in the backend and then connected it to the frontend results during this milestone. The reviews tell us a lot about how people feel about the show so out of all the similarity scores that factor into the tv show similarity results, we give reviews the highest rating. Also, the final app gives the user the choice to rate each result relevant or irrelevant. This rating will then factor into the result of the search and improve the outputs for the next search.

Class concepts:

- Cosine Similarity
- Tf-idf weighting
- Jaccard Similarity (for a previous prototype)
- Inverted Index
- Rocchio
- Relevance feedback
- Edit Distance
- Tokenize
- Term-Document Matrix

list the concepts (or practices) that you learned in class and used in your project (e.g., TF-IDF, cosine, term-document matrix, Rocchio, SVD, etc.).

Qualitative Evaluation:

Example 1

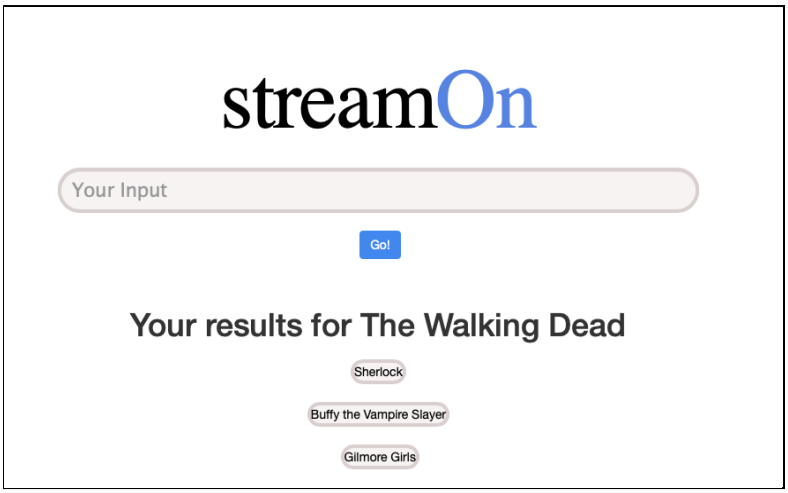
Input: TV show: The Walking Dead; **Similarity weight:** 100%

Output:

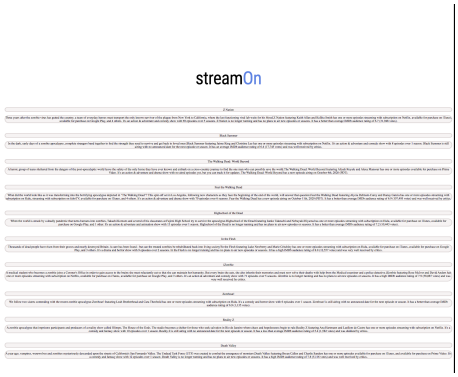
1. 'Fear the Walking Dead'
2. 'The Walking Dead: World Beyond'
3. 'Highschool of the Dead'
4. 'In the Flesh'
5. ‘Rubicon’
6. ‘Dead of Summer’'
7. 'Black Summer'
8. Z Nation
9. 'Freakish’
10. 'Mad Dogs'

Screenshot:

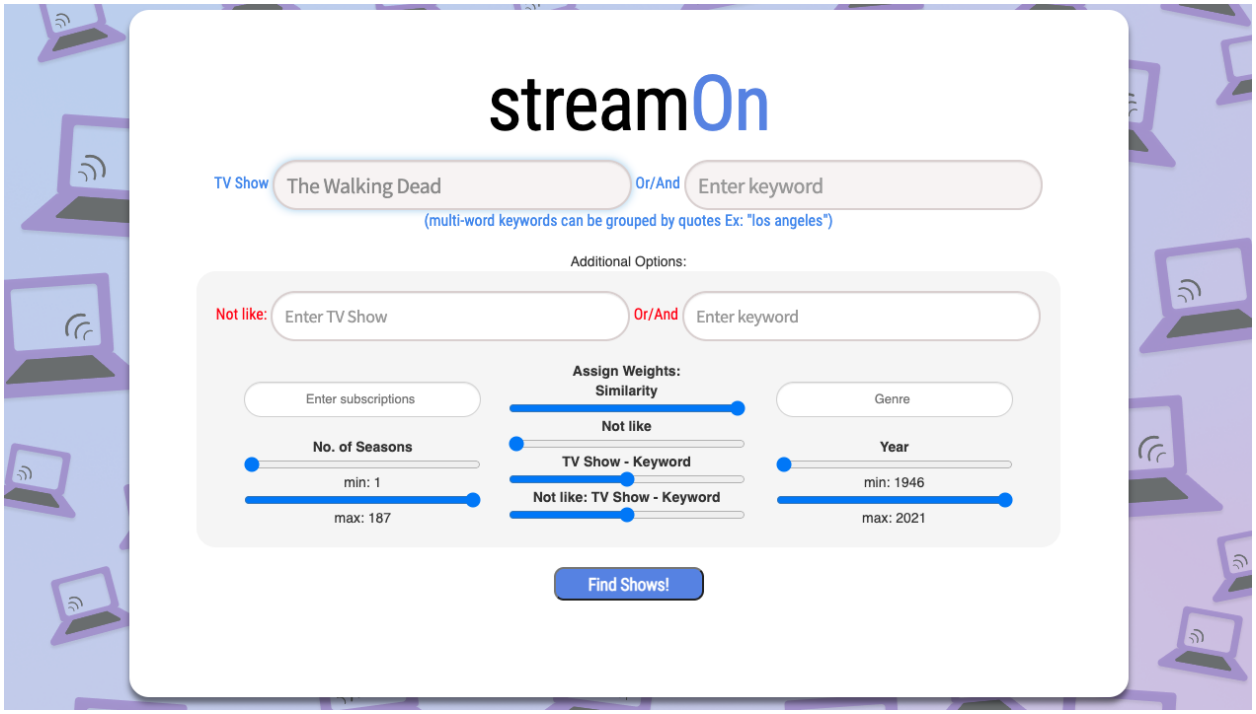
Prototype 1 (no weight)

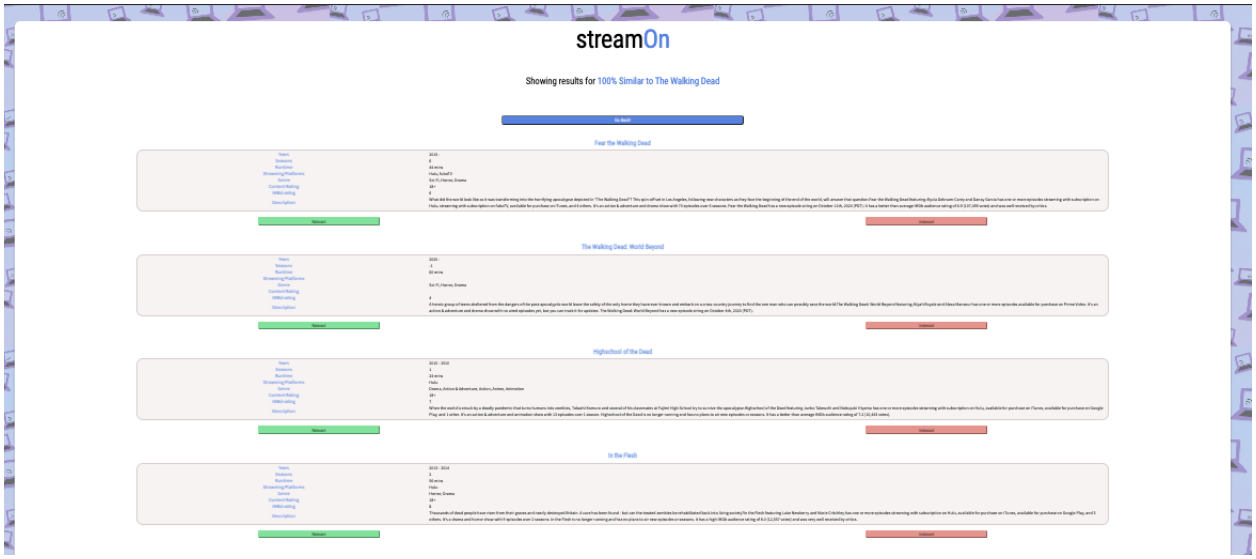


Prototype 2 (no weight)



Final App:





Example 2

Input: TV show: Sherlock; **TV show Weight:** 71%; **ad-hoc:** dogs; **ad-hoc Weight:** 29%; **Similarity Weight:** 94%

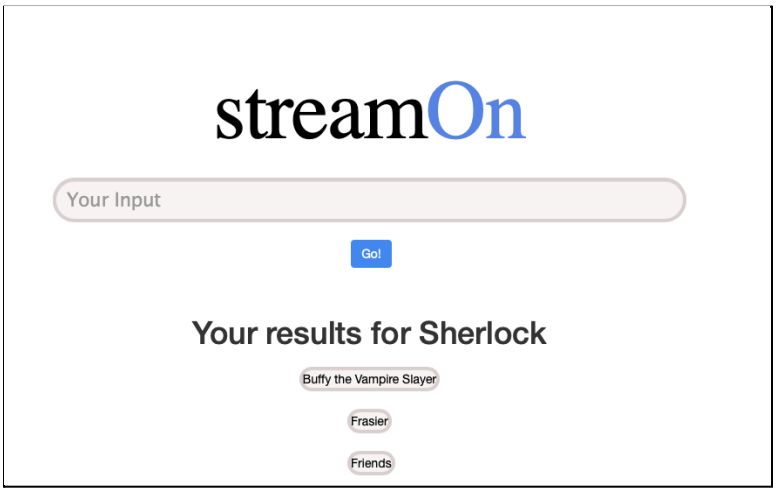
Output:

1. Harper's Island
2. Witchblade
3. Crossing Lines

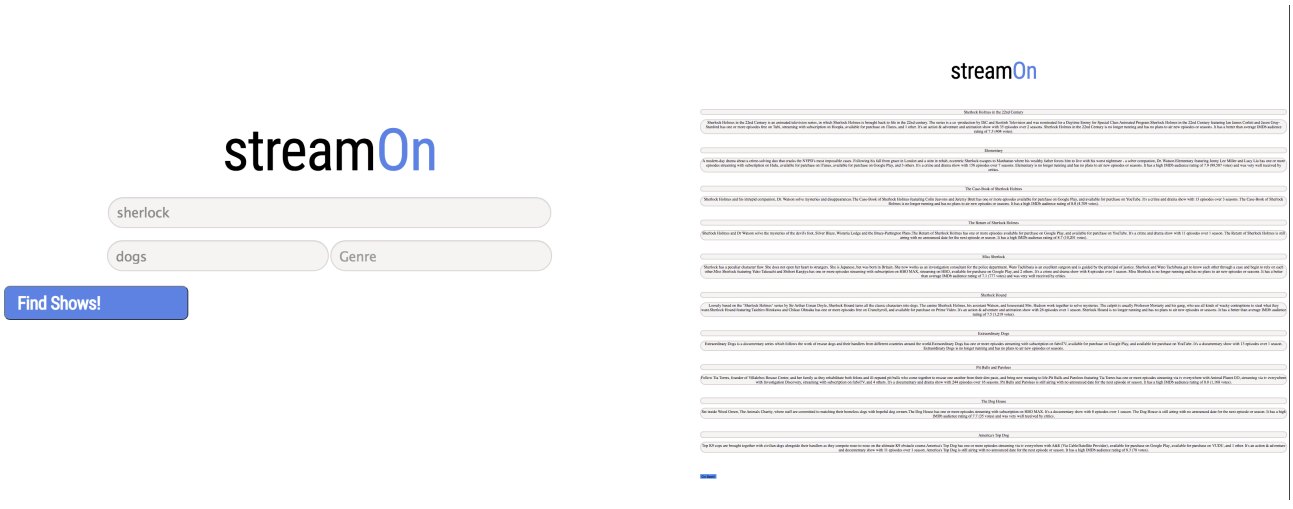
4. The Innocents
5. Glitch
6. Sherlock Hound
7. Pit Bulls and Paralees
8. The Dog House
9. The Name of the Rose
10. Master of None

Screenshot:

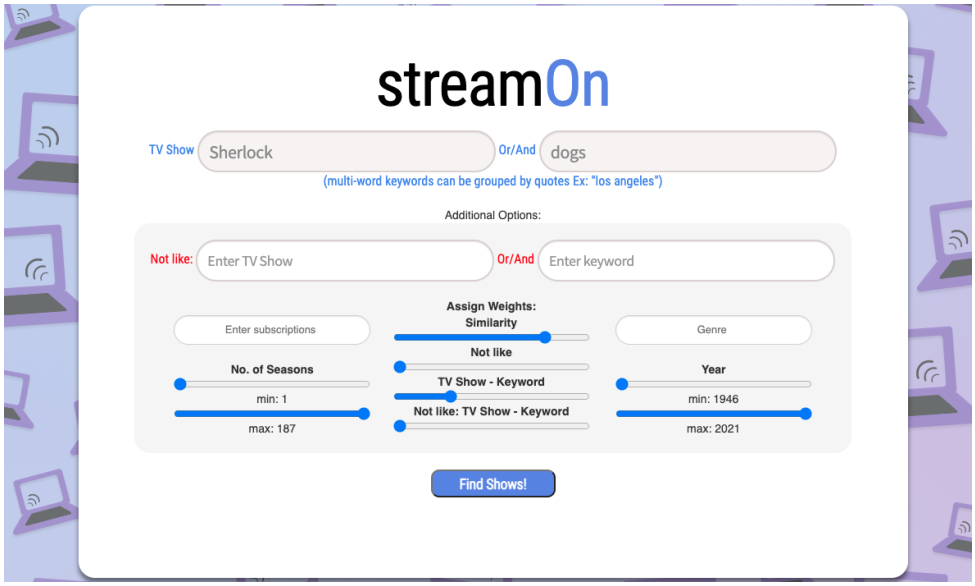
Prototype 1 (no ad-hoc component, no weights)

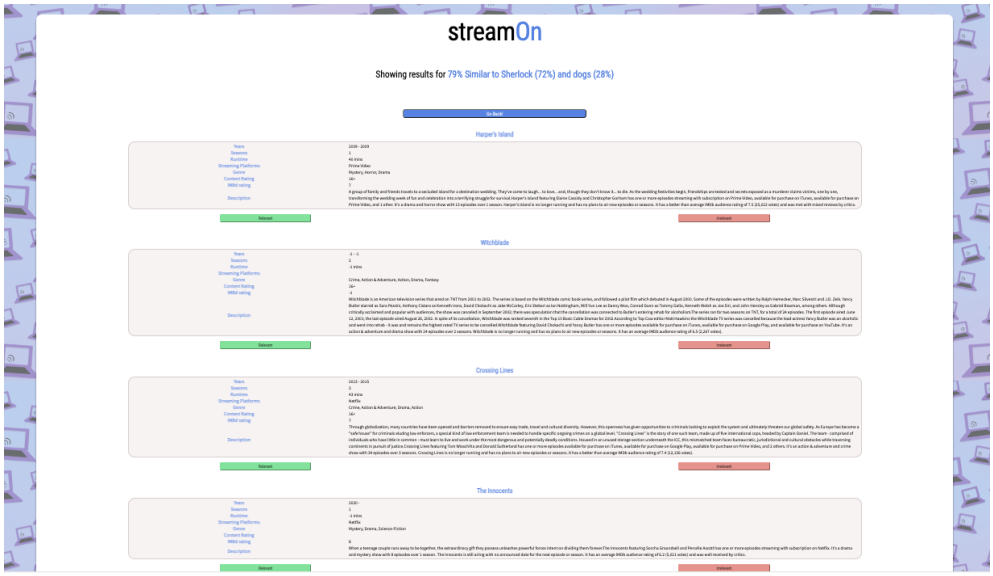


Prototype 2 (no weights)



Final App:





Example 3

Input: TV show: It's Always Sunny in Philadelphia; **Not Like TV show:** The League; **Similarity Weight:** 87%; **Not Like Weight:** 76%;

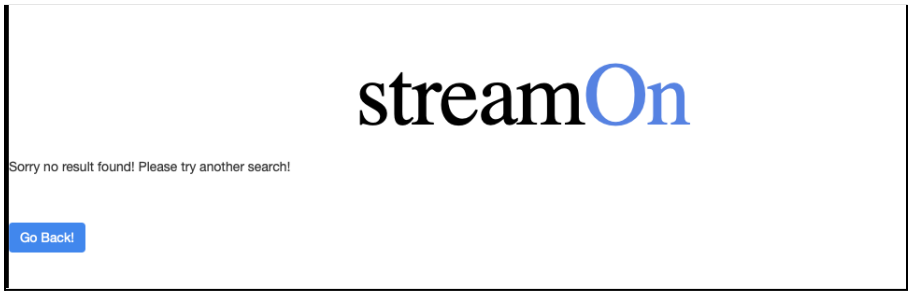
Output:

1. Transparent
2. Men at Work
3. The Last Man on Earth
4. Life on Mars
5. Rookie Blue

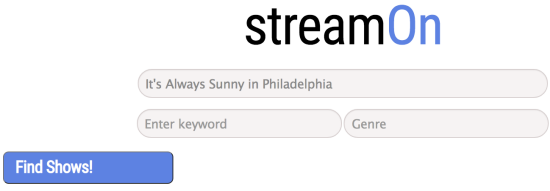
6. Cracked
7. The Loop
8. Mad Men
9. Stitchers
10. How to Get Away with Murder

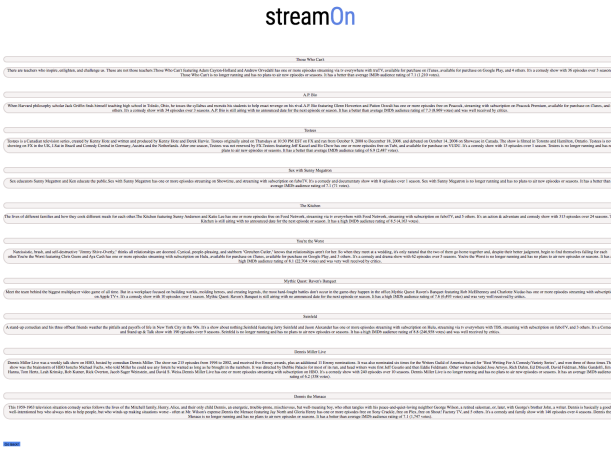
Screenshot:

Prototype 1

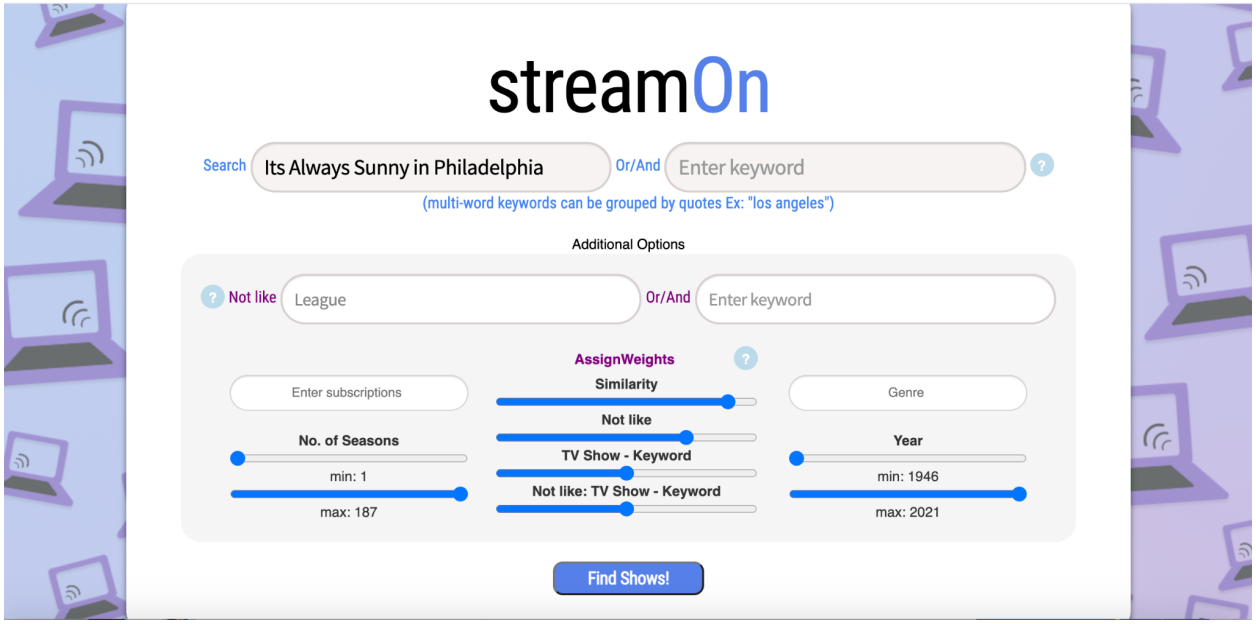


Prototype 2





Final App:



Example 4

Input: TV show: insecure; **ad-hoc:** “los angeles”; **Not Like TV show:** black-ish; **Similarity Weight:** 87%; **Not Like Weight:** 76%, **Streaming Platforms:** Netflix, HBO Max

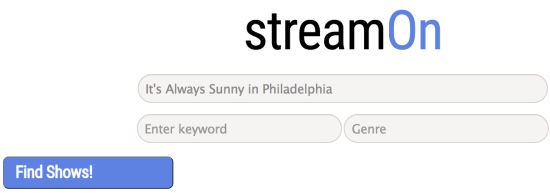
Output:

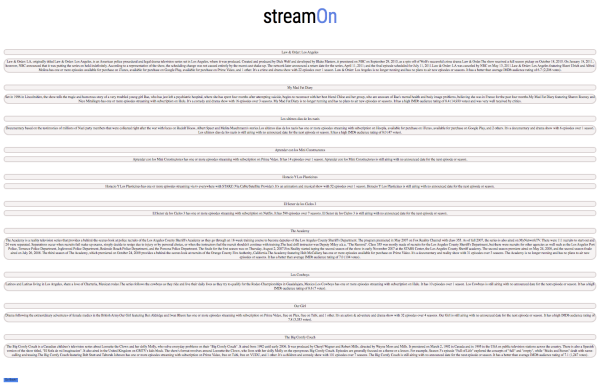
1. High Maintenance
2. Big Little Lies
3. Girls
4. True Detective
5. Game Of Thrones
6. Tell Me you Love Me
7. Euphoria
8. I May Destroy You
9. Sr. Ivila
10. On Freddie Roach

Screenshot:

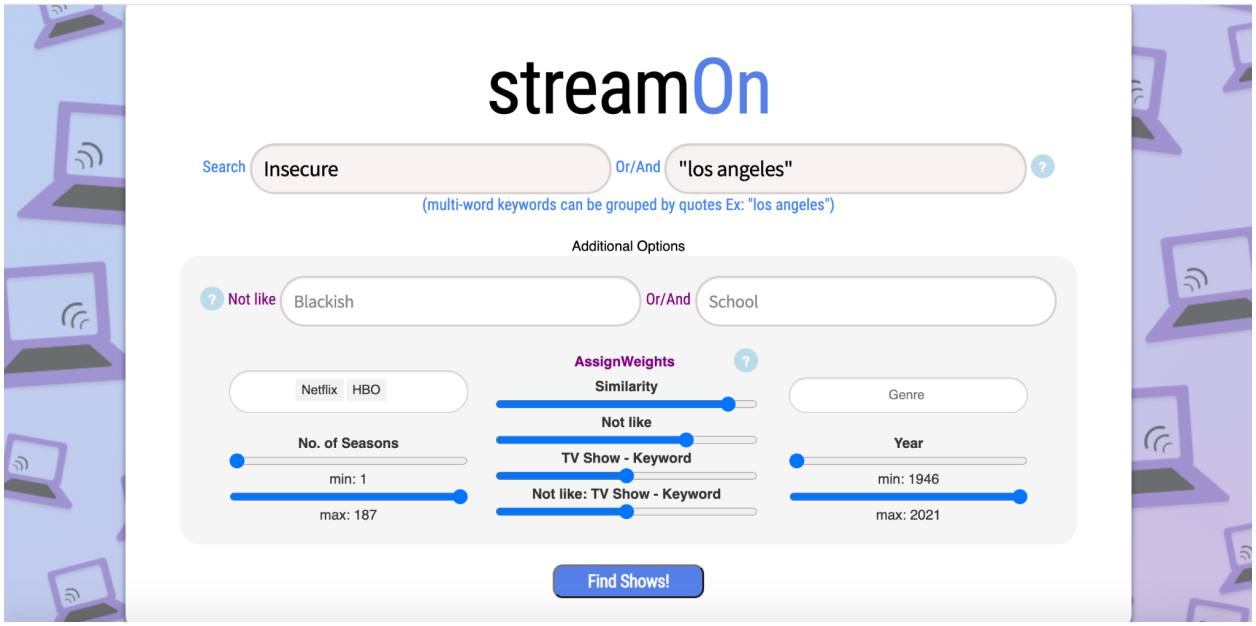
Prototype 1

Prototype 2





Final App:



Project Prototype (P04): Second running prototype

Description of changes and acknowledgment of feedback.

For the second prototype, we focused on adding our social and machine learning components to our app. In the previous model, on searching a TV show, the app would return the top three similar shows based on Jaccard Similarity of the transcripts. Now, the search results also take into account the Cosine Similarity of genre, the Cosine Similarity of descriptions, and the Cosine Similarity of reviews (from IMDb), each weighted appropriately to give the optimal results. We also improved our search by making the queries case insensitive and adding edit distance to account for misspellings.

We implemented relevance feedback for our machine learning component. When receiving results the user will indicate whether they found the document to be relevant or not. We will then store this information in a json file. If the percentage of users saying a result is relevant for a show is >0.75 then the result is added as a result doc for the show, and if the percentage is less than <0.25 then the result is added as an irrelevant doc for the show. We then use this dictionary of relevant and irrelevant docs for a show to power our rocchio update algorithm.

We also improved our UI by directing the results to a new webpage, improved the layout of search boxes, and providing appropriate error messages in case the TV show doesn't exist in our database.

Input Output Examples

Example 1

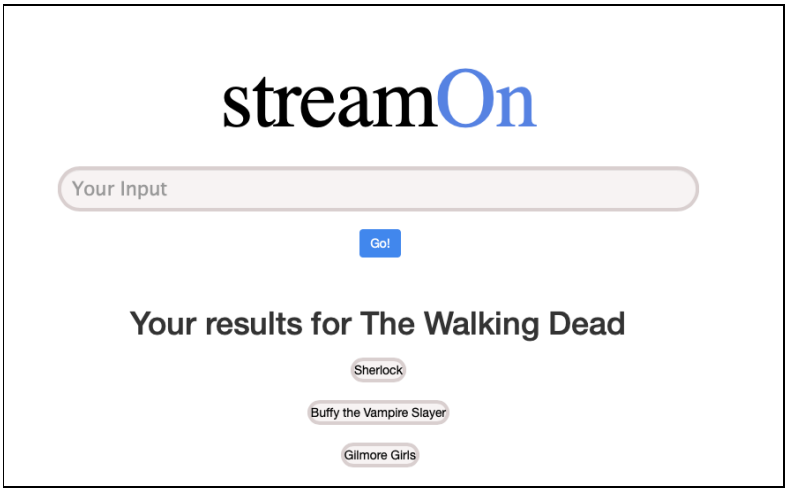
Input: TV show: The Walking Dead

Output:

11. 'Fear the Walking Dead'
12. 'The Walking Dead: World Beyond'
13. 'Highschool of the Dead'
14. 'Black Summer'
15. 'In the Flesh'
16. 'Z Nation'
17. 'Freakish'
18. 'Mad Dogs'
19. 'The 100'
20. 'The Returned'

Screenshot:

Prototype 1



Prototype 2

Description:
For this search, we entered one TV show "The Walking Dead" which gave the results based on the Jaccard Similarity of the transcripts, reviews and descriptions. While the results are accurate and as expected, we could look into providing an option to the users to determine how close the results should be to the input as someone who was seen "The Walking Dead" would most likely know about "Fear the Walking Dead" or "The Walking Dead: World Beyond". Overall, the results were accurate for our goals for this prototype.

Example 2

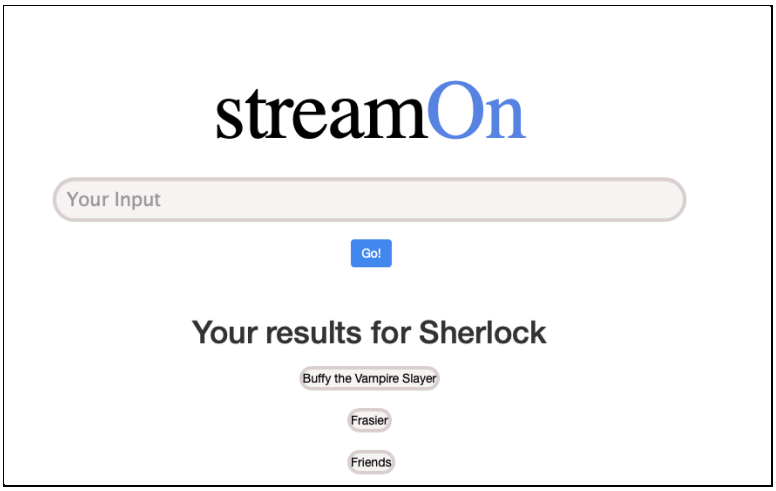
Input: TV show: Sherlock; **ad-hoc:** dogs

Output:

11. Elementary
12. Extraordinary Dogs
13. Pit Bulls and Parolees
14. The Dog House
15. America's Top Dog
16. Sherlock Holmes in the 22nd Century
17. Paul O'Grady: For the Love of Dogs
18. The Return of Sherlock Holmes
19. The Case-Book of Sherlock Holmes
20. Miss Sherlock

Screenshot:

Prototype 1 (no ad-hoc component)



Prototype 2

Description:
For this search, we tested our normal search with the show “Sherlock” and for the free search (ad hoc) we use the keywords “dogs”. For the base search, we used information from description, reviews, and transcripts. For the free search, we only used the reviews and descriptions. In our results, there were a lot of Sherlock shows and then dogs shows with similar themes. We might want to give less weight to ad hoc search inputs. Overall, the output worked really well.

Example 3

Input: It's Always Sunny in Philadelphia

Output:
none

Screenshot:

Prototype 1

Prototype 2

Description: For this search, we imputed the TV Show "It's Always Sunny in Philadelphia", but did not receive any results back. This is because our dataset currently does not have the transcripts, reviews or descriptions for this show. The user will be shown an appropriate message saying that the results weren't found and that they should try another search. Overall, the output was as expected for this show.

Example 4

Input: TV show: insecure; **Ad hoc:** Los Angeles

Output:

1. 'Law & Order: Los Angeles'
2. 'Master of None'
3. 'Mad Dogs'
4. 'Tell Me You Love Me'
5. 'Sense8'
6. 'Easy'
7. 'Girlfriends'
8. 'Swingtown'
9. 'The Neighbors'
10. 'American Crime'

Screenshot:

Prototype 1 (no ad-hoc component)

Prototype 2

Description:
For this search, we entered the TV show "insecure" which tests our basic search using transcripts and IMDb descriptions through Jaccard Similarity. The ad hoc query, "Los Angeles" tests our Jaccard Similarity for reviews from IMDb, which form the basis of our ad hoc component. As a result, there are a mix of shows that contain the keyword "Los Angeles", and shows related to "insecure". This search also shows that our inputs are case insensitive, i.e.

"insensitive" and "Insensitive" will both return the same results. Overall, the results came out as expected.

Example 5

Input: TV show: power

Output:

1. 'Mad Dogs'
2. 'Rubicon'
3. 'Game of Thrones'
4. 'Sense8'
5. 'Gomorrah'
6. 'Siberia'
7. 'The Returned'
8. 'The Tomorrow People'
9. 'The 100'
10. 'American Crime'

Screenshot:

Prototype 1

Prototype 2

Description:

For this search, we tested our normal search with the show “power” to test case-sensitivity . In our results, the suggestions were pretty accurate, but there were some outliers like “The 100” and “The Tomorrow People”. This might be because we have not added all of the transcripts into our prototype database. Overall, the output worked really well.

Project Prototype (P03): First running prototype

Prototype link: https://stream-on.herokuapp.com

Baseline Method Description

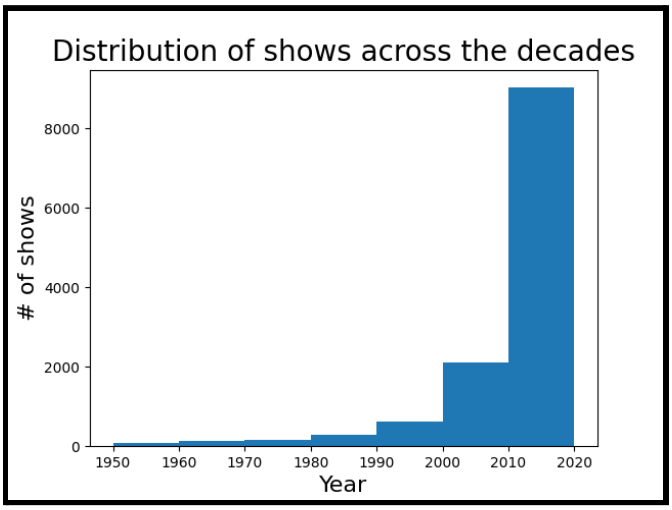
Briefly explain the baseline method that is currently generating output. What information retrieval and machine learning techniques (if any) are you using in your output? Does your method require any special data manipulation or post-processing?

For the baseline model, we focused on the main information retrieval aspect, which is to generate a ranked list of TV shows similar to the show the user inputs. For this model, we output the 3 most similar shows based on the Jaccard similarity between their transcripts. The model takes into consideration all words in the transcripts of a show that appear more than once per episode. It then uses the Jaccard similarity measure to rank which shows are the most relevant and returns the top 3 to the user.

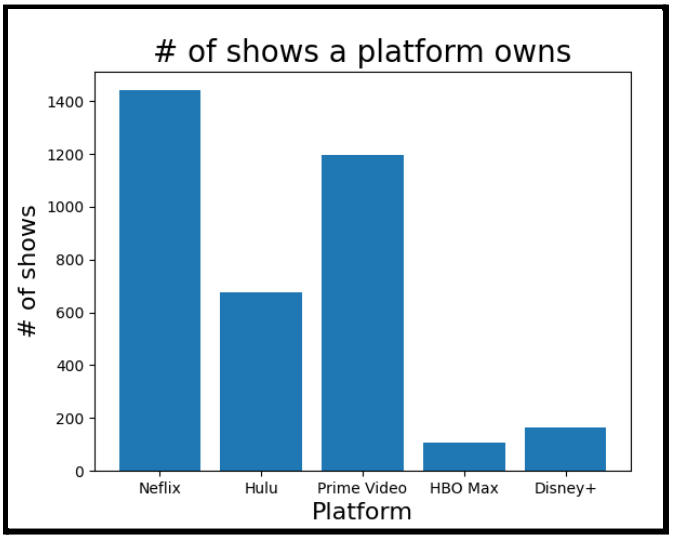
Project Update (P02): Data, Examples and Updated Description

Statistics and Basic Exploration of the Dataset

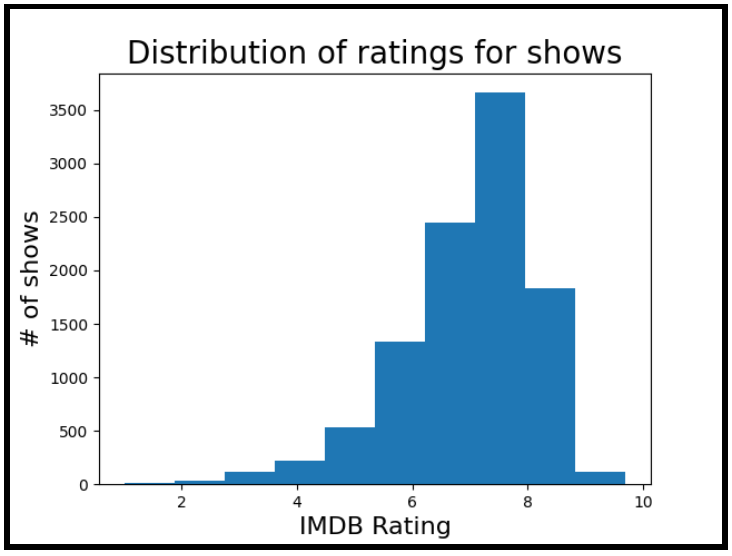
Our current dataset contains around 12450 shows. We believe that is an ample amount of data required for this project. Looking at the histogram below we see that most of the shows we will recommend come from the 21st century. This makes sense because most older shows either are not on many streaming platforms or our target audience would not care to watch.



There are a lot of ways to watch shows currently. With the rise of television streaming services and the diversification of available platforms, many shows are found on one or multiple different platforms. The shows in our dataset can be found on 5 major different platforms. The platforms that the most shows can be watched on are Netflix while HBO Max only has a few shows. This part of the data will be important in our recommendation, since we want to make sure we are able to have a significant number of recommended shows based on the input show and the desired platform that the user provides. Given that most of the platforms in our dataset have many shows, our recommendation can be reasonably accurate. However, we may struggle to provide multiple recommendations for HBO and Disney+, which are represented less in our dataset.



When ranking the recommendations, we want to provide shows that are not only highly relevant based on the input information, but also highly rated. It would not be useful to the user to provide show recommendations that are similar to their input but poorly rated and unpleasant to watch. We will have to find a balance between the show's rating and similarity. The distribution of ratings in our dataset is skewed higher on the scale, with most shows falling between the scores of 6 and 8. This will make it possible for us to provide high-quality recommendations.



Sketched Examples and Descriptions of Idealized Inputs and Outputs

For our TV recommendation system, the app will require the user to input the name of a TV show that they want the recommendation to be based on. The user also has the option to provide other information, such as additional TV shows that they want to include or exclude from the recommendation input, streaming platforms that they have access to, average desire episode length. show length, genres they want to include or exclude from the recommendation input, and more. The app will output a ranking of similar TV shows along with the streaming platforms where you can watch the shows. Each show in the list will contain a small description, year, avg. episode length, rating, and possibly a small graph to show how seasons change in similarity. Below are two examples of valid inputs and their outputs.

~ **User input:** American Horror Story

~ **System output:**

Here's a the top results for shows that are similar to **American Horror Story** (2011):

1. American Crime Story (2016) An anthology series centered around some of history's most famous criminals.American Crime Story featuring Darren Criss and Edgar Ramírez has one or more episodes streaming with subscription on Netflix, available for purchase on iTunes, available for purchase on Google Play, and 3 others. It's a biography and crime show with 19 episodes over 2 seasons (Avg Episode Length: 53 mins). American Crime Story has a new episode airing on September 27th, 2020 (PDT). It has a high IMDb audience rating of 8.4 (77,103 votes), Rotten Tomatoes Score of 93%, and was very well received by critics.

2. The Haunting (2018) - The Crains, a fractured family, confront haunting memories of their old home and the terrifying events that drove them from it.The Haunting featuring Michiel Huisman and Elizabeth Reaser has one or more episodes streaming with subscription on Netflix, available for purchase on Google Play, available for purchase on Prime Video, and 2 others. It's a drama and horror show with 10 episodes over 1 season (Avg Episode Length: 56 mins). The Haunting is still airing with no announced date for the next episode or season. It has a high IMDb audience rating of 8.7 (154,025 votes) and was very well received by critics.

3. Black Mirror (2011) - A contemporary British re-working of The Twilight Zone with stories that tap into the collective unease about our modern world. Over the last ten years, technology has transformed almost every aspect of our lives before we've had time to stop and question it. In every home; on every desk; in every palm - a plasma screen; a monitor; a smartphone - a black mirror of our 21st Century existence. Black Mirror has one or more episodes streaming with subscription on Netflix, and available for purchase on iTunes. It's a drama and fantasy show with 22 episodes over 5 seasons. Black Mirror is still airing with no announced date for the next episode or season. It has a high IMDb audience rating of 8.8 (417,642 votes), Rotten Tomatoes Score of 83%, and was well received by critics.

   ...

~ Given the query "American Horror Story" the system queries the database of tv shows that are similar to *American Horror Story*, which would include shows that are Horror or Dramas and Anthology Series. Then, we will precompute to ensure that the user has input a valid query. To determine similarity, we will compare transcripts, genre, descriptions, and ratings. We have a lot of data to search through so we will have to figure out the order of how we will narrow down the dataset and which factors have a higher ranking in terms of similarity. We need to decide if we want to show just the top 10 results or pages of results ordered by rank.

~ **User input:** Friends NOT The Office STREAMING SERVICES: Hulu, Netflix EPISODE LENGTH: <30 mins

~ **System output:**

Here's are the top results for shows that are similar to **Friends** (1994) but not similar to **The Office** (2005) on **Hulu and Netflix** with an episode length **<30 mins:**

1. Seinfeld (1989) - A stand-up comedian and his three offbeat friends weather the pitfalls and payoffs of life in New York City in the '90s. It's a show about nothing.Seinfeld featuring Jerry Seinfeld and Jason Alexander has one or more episodes streaming with subscription on Hulu, streaming via tv everywhere with TBS, streaming with subscription on fuboTV, and 3 others. It's a Comedy and Stand up & Talk show with 190 episodes over 9 seasons (Avg Episode Length: 22 mins). Seinfeld is no longer running and has no plans to air new episodes or seasons. It has a high IMDb audience rating of 8.8 (246,958 votes) and was well received by critics.

2. How I Met Your Mother (2005) - A father recounts to his children - through a series of flashbacks - the journey he and his four best friends took leading up to him meeting their mother. How I Met Your Mother featuring Cristin Milioti and Neil Patrick Harris has one or more episodes streaming with subscription on Hulu. It's a comedy and romance show with 209 episodes over 9 seasons (Avg Episode Length: 22 mins). How I Met Your Mother is no longer running and has no plans to air new episodes or seasons. It has a high IMDb audience rating of 8.3 (584,275 votes), Rotten Tomatoes Score of 83%, and was well received by critics.

3. New Girl (2011) - Jessica Day is an offbeat and adorable girl in her late 20s who, after a bad breakup, moves in with three single guys. Goofy, positive, vulnerable and honest to a fault, Jess has faith in people, even when she shouldn't. Although she's dorky and awkward, she's comfortable in her own skin. More prone to friendships with women, she's not used to hanging with the boys—especially at home. New Girl featuring Zooey Deschanel and Jake Johnson has one or more episodes streaming with subscription on Netflix, streaming via tv everywhere with TBS. It's a comedy show with 146 episodes over 7 seasons (Avg Episode Length: 21-24 mins). New Girl is no longer running and has no plans to air new episodes or seasons. It has a high IMDb audience rating of 7.7 (182,534 votes), a Rotten Tomatoes score of 94%, and was very well received by critics.

   …

~ Given the query "Friends NOT The Office STREAMING SERVICES: Hulu, Netflix EPISODE LENGTH: <30 mins," the system queries the database of tv shows with a runtime of 30 mins that are similar to *Friends*, which would include shows that are romantic comedies revolving

around a friend group, and not similar to The Office, which would include workplace comedies, and are on Hulu or Netflix. After making sure that it's a valid query, the algorithm will work like the above example except it will prioritize the input filters over the other filters that are already a part of the algorithm.

An Updated Description

We added new things to our abstract by updating our piazza post. The italicized parts represent the differences from P01 to P02. We added that one of the filters would be streaming platforms the user has access to because we had talked about it when brainstorming this idea and we think it will make the output more personalized. We also added a Creativity section because after meeting with our TA, we have begun thinking about what would make our system stand out. Although there are other platforms that recommend TV shows, our platform has two distinct features. First, our system is entirely dedicated to TV shows on streaming services, so it only makes sense that it recommends shows that the user has access to. Second, we want to have an option to recommend a show to a group of people. If the users are a group of people, multiple people will fill out their preferences and then output movies that relate to as many of the group's preferences as possible.

Weekly Schedule

Week 4/4
- Project Manager: Kendall
- Organize Datasets
  - Team Members: Kendall, Riya, Divya
- Data Exploration
  - Team Members: Sid, Siddhi

Week 4/11
- Project Manager: Divya
- Finalize Datasets
- Begin Working on Implementation
- have all the basic recommendation working

Week 4/18
- Project Manager: Riya
- Work On Implementation
- include "mood" and user ratings
- Work on UI
  - Team Members: Siddhi, Kendall, Riya
- Prepare Demo
- **April 19: First live prototype, a presentation and demo in class**

Week 4/25
- Project Manager: Siddhi
- Finish Implementation
  - Team Members: Everyone
- Finish UI
  - Team Members: Siddhi, Kendall, Riya
- Testing
- Prepare Demo
- Team Members: Everyone
- **April 30: Second live prototype, presentation**

Week 5/2
- Project Manager: Sid
- Finalize Everything
- Prototype / Testing
- Presentation
- **May 5: Final app goes live, presentation**

Experience/Work Division

Siddhi:

- Prefer to work on backend work, can help with frontend

Divya:

- Backend - highest preference
- Machine Learning (Moderate experience - some NLP (little experience with summarization) and other more generic models like SVMs, KNN, Decision Trees, etc.)

Riya:

- primarily Backend
- Design (UI) by working with Frontend
- can help out with NLP (moderate experience)

Kendall:

- Prefers Backend
- Has experience with Frontend

Sid:

- Prefers Backend

Project Proposal Abstract (P01)

Piazza Note: https://piazza.com/class/kkphiirsh2a8p?cid=487
GitHub Repository: github.com/riyajaggi/cs4300sp2020-rj356-dd492-sc2538-sv352-kal255
Heruko App: https://stream-on.herokuapp.com

Goal

The app will recommend similar TV shows based on user input and will tell the user which platform the show can be streamed on. This can be extended to also account for episode length, moods, and can utilize summaries of the TV shows.

Description

Input: The app will require the user to input the name of a TV show that they want the recommendation to be based on. The user also has the option to provide other information, such as additional TV shows that they want to include or exclude from the recommendation input, streaming platforms that they have access to, average desire episode length and show length, genres they want to include or exclude from the recommendation input, desired age rating, and the range of years they want the movie to be from. The app can also be extended to provide results based on mood categories that the user wants the output to fall into.

Output: The app will output a list of similar TV shows along with the streaming platforms where you can watch the shows. Each show in the list will contain a small description, year, avg. episode length, rating, and possibly a small graph to show how seasons change in similarity. Along with this info the output will contain the ranking of the list. Also it will ask the user for feedback to check if the output is akin to what the user expected or wanted.

Use Cases: This app would be used to find similar (or dissimilar tv shows) given an input, and tell the user what platforms they are available on. The user will also be able to find shows based on their mood (happy, sad, thoughtful, scary, etc). Also by using additional TV shows option, they can find recommendations for group viewing.

Data Sources

- TV shows on Netflix, Prime Video, Hulu and Disney+ (Kaggle): https://www.kaggle.com/ruchi798/tv-shows-on-netflix-prime-video-hulu-and-disney
- TV Show Transcripts: https://tvshowtranscripts.ourboard.org/viewforum.php?f=139
- Web Series: Ultimate Collection (Kaggle): https://www.kaggle.com/amritvirsinghx/web-series-ultimate-edition
- IMDb Datasets: https://www.imdb.com/interfaces/
- The TV DB: https://thetvdb.com
- Web Scraper for IMBd and Rotten Tomatoes: https://github.com/masumrumi/imdb_rotten_web-scraping

Information Retrieval

The app will search for existing TV shows and their meta information like genre, age rating, reviews, year, availability, etc from the various data sources. The query will take in the name of the TV show the user wants the results of and other preferences like the genre or the length. The results will display a ranked list of similar TV shows with the streaming platforms they are available on.

Social Aspect

The app will look at the reviews posted by the previous viewers on platforms like Rotten Tomatoes, Google Reviews etc which would help us to better understand the moods and sentiments of the different shows and help us provide more personalized recommendations.

Machine Learning (optional)

We are looking to make the search more complex than just entering one TV show. We would like to allow users to add in multiple shows, genres, avg. episode length, etc to get more personalized recommendations. We are also thinking of adding in "moods" and "summaries" which could require more complex traversing of dataset and some machine learning aspect. ML could also come into play when weighting all of these inputs, when finding the results. Reviews and ratings could also help us with "sentiment analysis" and cope up with any missing data from datasets.