Introduction

Netflix is one of the biggest streaming platforms in the world and offers streaming for thousands of movies and shows. In order to allow its users to find content that they will enjoy, it employs a complex recommendation system that takes many factors into account, such as the popularity of content and a user's own tastes. For our final project, our group wants to try our hands at creating a recommendation system, similar to what we learnt in class, in order to gain insight into the considerations behind Netflix's recommendation and how an effective recommendation system operates. Our goal is to look at data from Netflix, such as the rating and genre of shows and films offered by the service, and look into creating a content-based recommendation system through the use of cosine similarity.

In order to do this, we are making two iterations of our recommendation system. The first iteration will be our test run, so to speak. We will test this initial system and give recommendations to our peers using it. Then, we will have them do a feedback survey in order to gain insight on how to improve our model. From here we will make a second iteration of a recommendation system. We will show the results of this version to the same peers in order to see where we have improved and where we could go next.

About the Data

Our data comes from an online Kaggle dataset called [Netflix Chronicles: Exploring Movies and TV Shows]() by Nayana CK. The dataset contains over 8000 rows, reach containing data on a movie or show on Netflix. There are 11 columns of data, each of which gives information about the item. This includes an ID for the show, its type, the title, the director, cast, country of production, the date it was added to Netflix, the release year, its rating, the duration, and the categories/genres of the movie/tv show. As a whole, the Netflix dataset offers a comprehensive collection of information on the streaming material that is accessible and the result of using useful resources for conducting research and investigating trends in the entertainment industry.
In detail, these are the columns:

- show_id: a unique identifier for each show
- type: Indicates whether the entry is a movie or a TV show.
- title: The title of the movie or TV show.
- director: The director(s) of the movie or TV show
    - There are many null values in this column.
- cast: The cast members of the movie or TV show.
    - There are many null values in this column.
- country: The country or countries where the movie or TV show was produced.
    - There are many null values in this column.
- date_added: The date when the movie or TV show was added to Netflix.
- release_year: The year when the movie or TV show was released.
- rating: The audience rating assigned to the movie or TV show, as in what audience was the show/movie geared towards.
- duration: The duration of the movie or TV show.

- listed_in: The categories or genres that the movie or TV show belongs to.
    - The genres/categories of the title are kept in a comma separated list.

Many of these rows contain helpful data, but others we did not believe were necessary for a successful recommendation system. One concern is many of these rows contain textual data, which will not work with cosine similarity.

Methods:

Model 1:

Data Preprocessing:

To clean the data, we first dropped any columns of irrelevant data for our model. First, we dropped the director, cast, and country columns. This is because the data in these wasn't important for our model. Another reason was because these features included a substantial number of null entries. The director column alone had over 2500 null rows, which is too much to drop completely. Trying to one-hot encode this data is also unrealistic, as there are so many possible values that it would significantly increase overhead.. In order to simplify the dataset and get rid of columns that would not have a substantial influence on the recommendation system, this choice was taken.

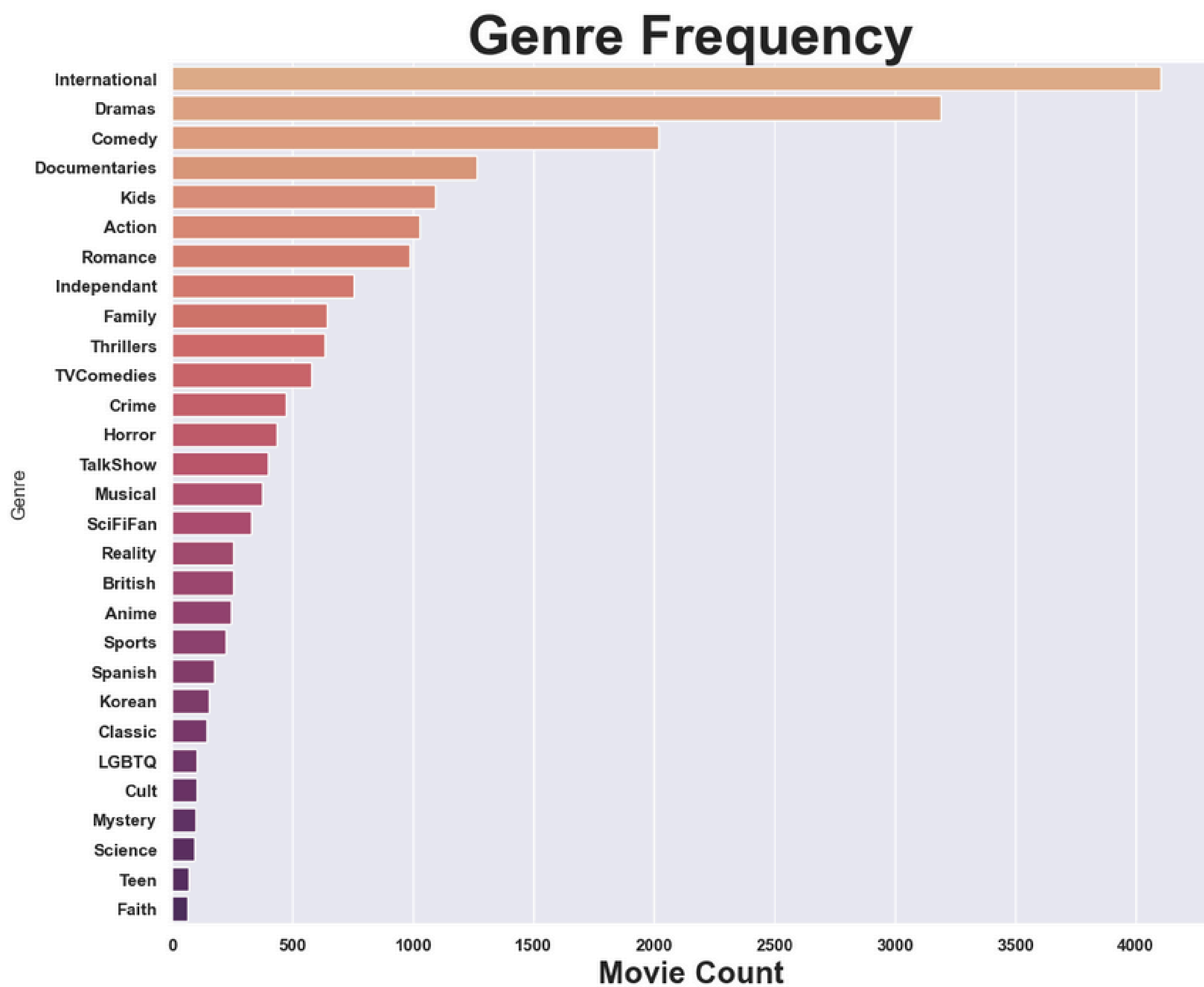
Duration was also dropped. Looking at the data in this feature, we saw that two different metrics were used in order to measure duration. Shows used season length while movies used minutes. This disconnect made it tough to use. The last column dropped was the date_added column, and this was just done because we were unable to think of a solid reason as to why it would be relevant.

Next we renamed some columns, solely for the sake of clarity and ease when coding. Release_year was changed to year, listed_in to category, and show_id to id.

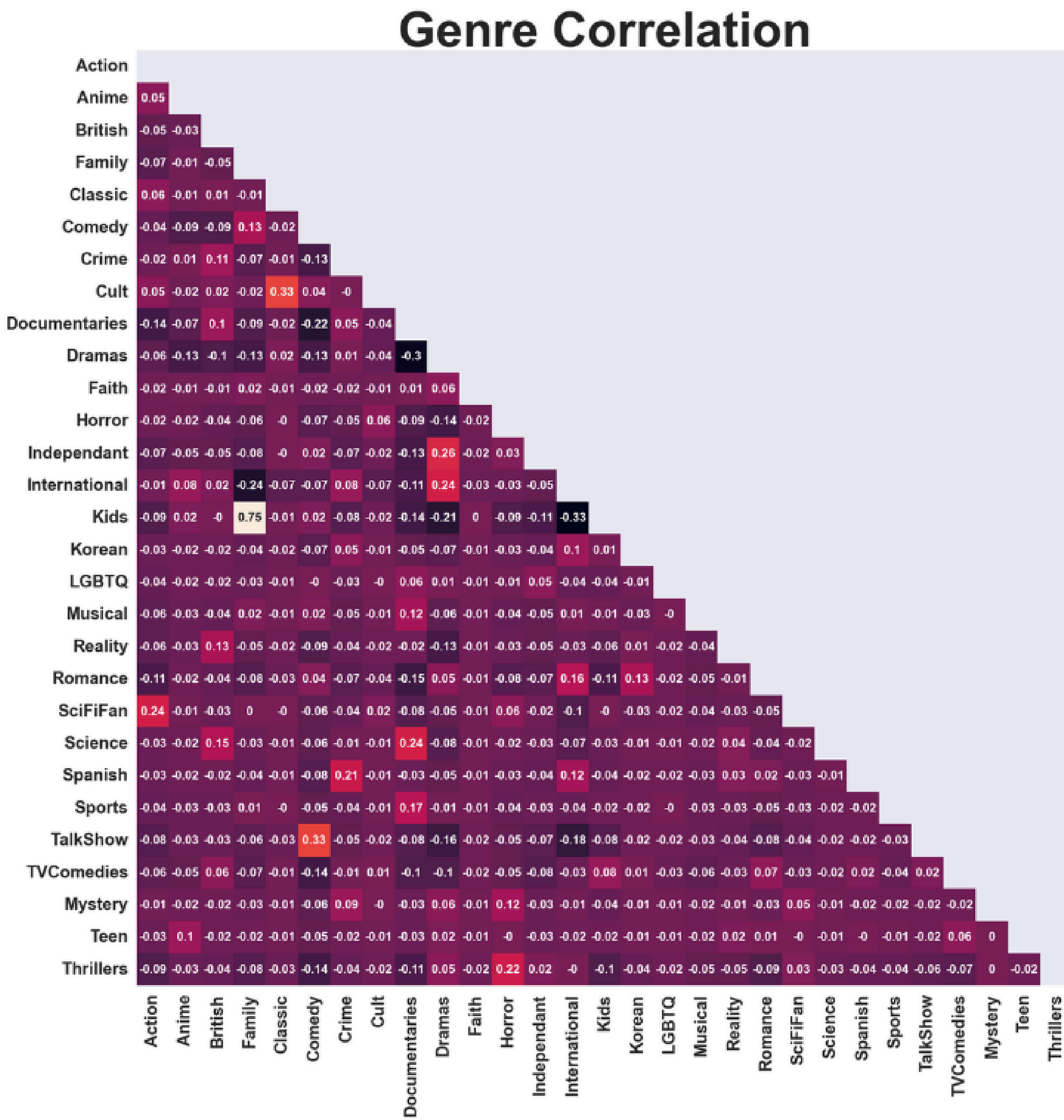
After this step was hot-encoding our categorical data in order to finish preparing it for the model. This was the most involved step, as while it was simple to separate the comma-separated lists, there was a lot of overlap between categories that we sought to fix. A lot were separated into genre and type both (ex/ comedy movie vs comedy show), which we found redundant. In order to fix this, we manually went through all the columns and merged any two that were similar. Some examples of this were merging anime features and series, dramas and tv dramas, and horror movies and tv horror. By one-hot encoding our data this way, it is now possible to use it for cosine similarities, and therefore recommendations. Alongside our merging, we also dropped some features. These were the TV and Movie categories, since we found it irrelevant at this point.

As one last step, we changed some of the category names for clarity. For example, we changed Children&FamilyMovies to simply Family.

In order to better understand the genre spread we created, we made some visualizations:



Through this bar graph, we can see that international films and shows make up by far the largest percentage of the films, while most of the other categories have significantly less data. This could cause concerns over not having enough data to go over when recommending shows/movies in these categories.



Looking at the overlap between genres can also give us some insight into what sort of genres of movies/shows will be recommended depending on our requests. For example, inputting a kids film is likely to return family shows as well. There are also areas of negative correlation, such as kids tv/movies and dramas. These also make sense, as certain genre combinations are not popular. Drama isn't considered suitable or interesting for children's media, so there isn't much overlap between the two. An interesting negative correlation exists between kids media and international media. I do not believe that this is due to a lack of content, but perhaps due to the way netflix categorizes media. For example, much of the anime in the dataframe are only labels as such, without any other genre attached to them to share whether it is a family anime or a horror anime. These oversights in categorization could significantly affect the results of our system.

Making the Model:

As referenced before hand, we used cosine similarity scores in order to create recommendations. We did this solely on the categories/genres, since we believed this to be the most important factor to make recommendations. As such, it represents the similarity between two title's genres.

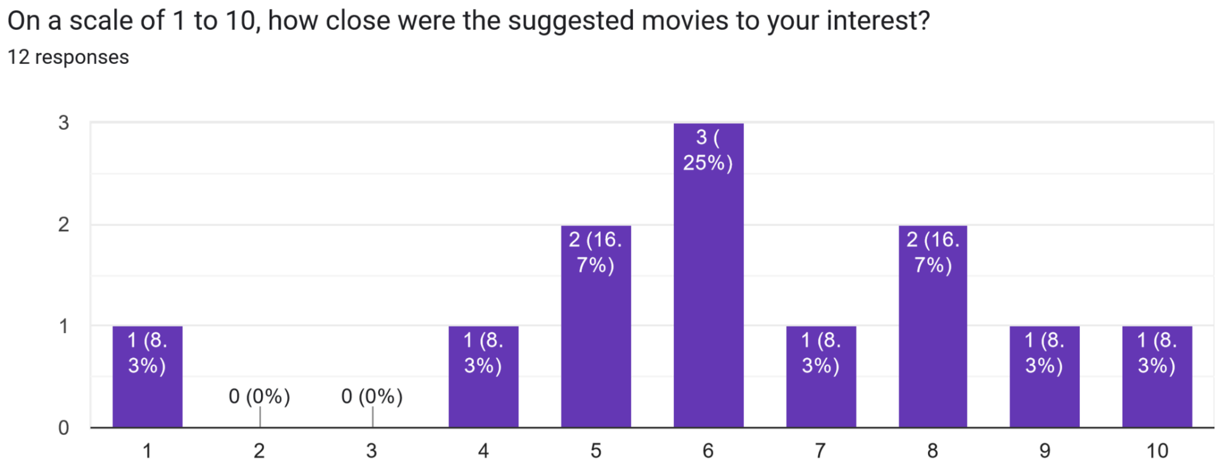
After calculating cosine similarity scores for all pairs of items in the dataset, the recommendation system can be created. We built a function that takes in the title of a movie/show, the cosine similarity matrix created using the netflix genre data, the original data, and the number of recommendations requested. From there the function checks that the title is in the dataset before continuing. After being found, the title is used to reference the related data in the similarity matrix. These results are used to find the top most similar titles in the dataset. From these top results, the requested amount of titles are returned back to the user as their recommendations.

And with that, we had our first iteration of a recommendation system

Evaluation:

In order to properly understand how well this system works, we must test it. In order to get a wide range of perspectives, we decided to run a survey. In this survey, we asked our peers to give us a movie or show that they enjoyed. We took that recommendation and ran it through our system, giving the results back to our peers. We then had them fill out a survey over how they felt over the recommendations. We asked them some various questions in order to gather both quantitative and qualitative data. With the quantitative data, we could have a better grasp of if we were improving. However, the importance of the qualitative data cannot be understated. This is how we got feedback over where we were succeeding and failing, allowing us to understand how to improve our second iteration.

The feedback we received was generally positive, with the majority of our peers feeling that the model was accurate to some degree.



While this was reassuring, we must also consider any biases that our peers may have. At the front of these concerns is that our peers are our peers, and as such are likely to give us results in our favor. This favor becomes apparent when running the survey. Frankely, many of the recommendations our program gave were incredibly inaccurate, or seemed completely out of the blue. Children's show recommendations were giving mature and horror adult show recommendations. There was a thematic disconnect between the films being suggested. Since "Anime" was its own isolated category, whenever you put in one, oftentimes you would get a random list of other animes with no regard for the details of the show.

The qualitative data seemed to agree with our observations, with one participant saying "it seems to make the recommendations based purely on somewhat vague and open-ended genres". This made it very clear to us that genre was not enough to make a decent recommendation system.

Model 2:

Data Preprocessing:

In order to improve our next iteration, we decided to one-hot encode more categorical features. Looking at all the other features, we decided that "type" and "rating" would be the best features to add in order to improve our results.

Type was added after considering the sort of recommendations we expected users to want when inputting a title. For example, if you put in a movie, you are likely looking for another movie to watch. Since there were only two categories (Movie or TV Show), this was very easy to one-hot encode into our cosine matrix dataframe.

We also came to the conclusion that the prevalent issue of iteration one, of recommendations being geared towards vastly different audiences, could be solved by using the rating. Getting only G rated films when inputting a G rated film, or only rated R shows when inputting as such would limit the amount of "cross-contamination" we were having. One-hot encoding this data was more complicated however, as the data was messier. Some of the ratings were actually times, and we had to drop those ratings from our data. Otherwise, it was mostly a matter of renaming columns in order to be better understood.

By adding these two features to the dataframe, we were able to hopefully create a significantly more accurate recommendation system.

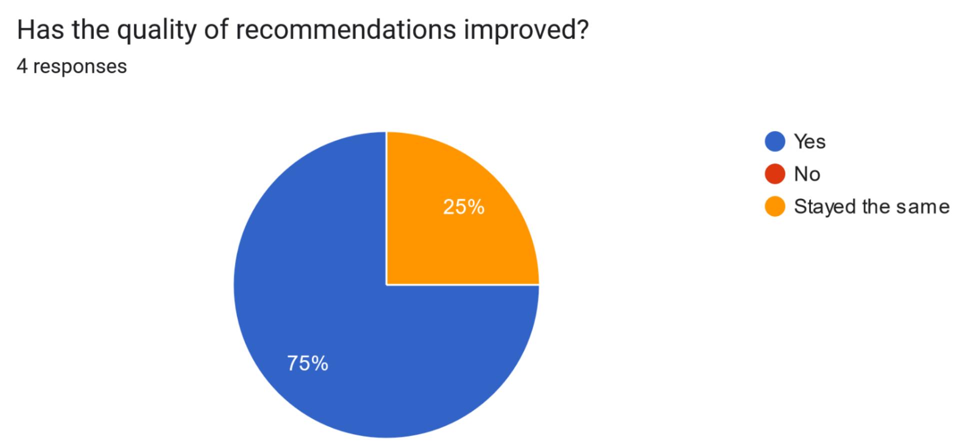
Making the Model:

This model was made very similarly to our first iteration, with the major difference being the data inputted into sklearn's cosine similarity function having added rating and type features.

There were also differences in our implementation of the recommendation function. This time, we designed it such that it could take in a list of films or shows. It is able to check all of the titles inputted against the cosine similarity matrix and average the results. By averaging the results, we are able to find a middle ground between all of the users interests. By taking in a larger amount of data from the user, we are hoping for significantly more accurate results.

Evaluation:

In order to measure if we have improved our system, we ran another survey to get our participants' thoughts. In order to get the most standardized results, we ran the survey on the same people. Instead of taking in new titles and recommending them, we took the same titles that they asked for in the initial survey and ran those through our second iteration. This way, the participants had a reliable way of comparing whether or not the system had improved. Looking at the results, we get the following visualization:



While we were unable to get the same amount of responses (due to many of our peers being busy with the chaos that comes with the end of the semester), we are able to see that the feedback is positive. While perhaps not a significant amount of responses to come to a reliable conclusion, it is als something we noticed when running the survey. This version gave more reliable results, with all titles being aimed at similar audiences and covering similar topics. We credit this mostly towards the combination of using category and rating features in order to gauge the similarity between two features. From the results we got, it looks like they agreed. One participant remarked that "the quality of the recommendations have improved, as it shows me movies of both animated and live action with similar themes."

With these results, we are confident that there has been a marked improvement in our model.

Conclusion:

Through this project we were able to better understand the considerations that go into building a recommendation system. While we wouldn't call our recommendation system perfect, it was a good introduction into the subject, and allowed us to learn many of the considerations that go into feature selection for these systems. Using just a single metric to gauge similarity is flawed and will lead to wildly out of touch recommendations.

Even with our changes, we barely even scratched the surface. We threw around the idea of using the description as another marker of how similar two titles in the dataset were, but ultimately were at a loss as to how to properly implement such a thing. Some other complaints that were common in our survey had to do with the amount of data we had in the dataset. Many times, we had to get several titles from our peers before we could give them any recommendations. This problem may be in part to the proportion of international titles in the dataset we used. As international media isn't as popular in the US, or at least for our peers, finding a hit was harder.

This project—and class— taught us a lot about how to understand and evaluate problems in our projects. We were able to look at a problem, consider how to solve it, and go through the process of experimenting with our ideas. From there, we had to evaluate our results and troubleshoot how to improve them. This came with thinking outside of the box and considering many factors.

Impact:

Enhanced User Experience:

There is the potential for a considerable improvement in the user experience on the Netflix platform with the adoption of a recommendation system that is based on genre similarities.

The provision of customized suggestions that are suitable for the preferences of individual users increases the likelihood that users will find information that is relevant with their interests, which ultimately results in higher engagement and pleasure.

Content Discovery and Diversity:

Users are able to find a wider variety of information that goes beyond their typical interests with the help of the recommendation system.

When consumers are presented with suggestions for film or television series based on the similarities in genre, they are given the opportunity to discover material from a variety of genres or cultural backgrounds that they would not otherwise explore. This encourages variety and inclusiveness in the consumption of media.

Optimized Content Curation:

When it comes to Netflix, the recommendation system has the potential to enhance the discovery of content for viewers and optimize the curation of material. In order to better understand audience tastes and to build a library of material that connects with its broad user base, Netflix is able to better identify user preferences and watch trends via the use of analytics. This has potential to result in greater user loyalty and retention.

Business Impact:

A more effective recommendation system can have a positive impact on Netflix's business metrics including the growth of its subscriber base, the engagement of its viewers, and the amount of income it generates.

Netflix is able to maintain its present members and attract new ones by providing tailored suggestions that keep consumers engaged, interested and pleased. This helps the company to drive growth and keep increasing its competitiveness in the market.

Ethical Considerations:

Recommendation systems raise ethical concerns, especially in relation to user privacy and the use of data. Netflix must prioritize openness and accountability in the collection, storage, and use of user data for the purpose of generating recommendations. In addition, the recommendation system should be specifically designed to reduce any biases and actively encourage the fair portrayal of a wide range of material.

Environmental Impact:

Although not directly linked to the recommendation system itself, the project might potentially have an environmental effect as a result of heightened streaming activity.

Streaming services have substantial energy consumption, which leads to the release of carbon emissions and contributes to environmental damage. Netflix and other streaming platforms must adopt strategies to reduce their impact on the environment, such as using sustainable energy sources and enhancing the energy efficiency of their data centers.

In general, despite the fact that the project has the potential to improve user experience and economic results, it is vital to take into account the large social, ethical, and environmental consequences and to take actions to reduce any negative effects that may occur.

All of our code can be viewed through GitHub