# Case Study Part 1

There is a tremendous amount of competition that is present in the video streaming market, and one of the fastest growing streaming services is Disney+. Many disney fans are absolutely stoked about the content that is available on this platform, and they definitely should be. Colloquially, Disney is known for producing high quality movies and TV shows, and there is plenty of data out there that can help to tell the story of what content is available on the service, and of course, how good thjat content is.

After browsing Kaggle for a while, I came across a csv file that contains (what appears to be) every piece of content that is available on Disney+ at the moment. See the reference list for the link.

The features that are available in this dataset are:

imdb\_id title plot type rated year released\_at added\_at runtime genre director writer actors language country awards metascore imdb\_rating imdb\_votes

While this dataset has a Kaggle Usablility Score of 9.4 out of 10, it still may need to be cleaned a bit to be able to work with pandas properly. I have found that for attrubutes with multiple values (such as genre), the value is a comma separated list of values. I need to find a way to split these apart into multiple values or something, or maybe just strip the first from the list. (perhaps the first one is really the most important?) Also, some values are objects by default, so I may need to convert them to numeric type, this includes the runtime, which mostly is in the form "x min".

I look forward to working with this data set more closely, including doing some text analysis of the plot. This field is a plaintext representation of the plot of every show, movie, or series on Disney+, so I wonder what patterns will show themselves.

### References:

Disney plus movies and tv shows. (n.d.). Retrieved April 19, 2020, from https://kaggle.com/unanimad/disney-plus-shows

# Case Study Part 2

For this part of the Case Study, I decided to try to parse some of the fields that had comma separated categorical values. For example, a Movie can be listed as being in more than one genre. I wanted to encode this list of categorical values as numeric, and in order to do that, I needed to label encode them. This involved the following piece of code:

Label encode the genres  
df['genre'] = df.genre.fillna('')  
df\_genre\_labeled = (df.genre.str.split('\s\*,\s\*', expand=True)  
 .stack()  
 .str.get\_dummies()  
 .sum(level=0))  
df[df\_genre\_labeled.columns] = df\_genre\_labeled

This code will convert the values to strings, and split them into a list delimited by commas. It then stacks the items, and get\_dummies will label encode them. Sum will make the values either 0 or 1. (If the sum was anything else, then I know there was some problem. Luckily there was not. I then added these new columns to the previous dataframe, df.

I then took this same approach and applied it to the language, countries, and actors fields as well. There was a section for director, but that field has some additional information in it, that would not do well to be subjected to the same process. Here is why:

The data sometimes looked like this:

John Lasseter, Rob Gibbs(co-director), Victor Navone(co-director)

I do not want to lose the information that the "(co-director)" indicates, so i have held off on label encoding that data for now. A similar problem was present with the writer field, where the data would look something like:

John Lasseter (original story by), Andrew Stanton (original story by), Joe Ranft (original story by), Andrew Stanton (screenplay by), Don McEnery (screenplay by), Bob Shaw (screenplay by)  
Jymn Magon (story), Jymn Magon (screenplay), Chris Matheson (screenplay), Brian Pimental (screenplay), Brian Pimental (story supervisor), Curtis Armstrong (additional written material), John Doolittle (additional written material)

I may end up simply removing this data, because I don't know what I will do with this additional level of detail.

I decided to visualize the categorical data using horizontal bar charts, sorted by frequency. I also decided to remove lines which were clearly erroneous, as they were very largely empty.

# Case Study Part 3

In the previous two sections of this case study, I started by cleaning the data as much as I could, and then tried to create some basic visualizations of the patterns that were present for the content on Disney+. The vast majority of the data that I had available was categorical rather than numeric, so I needed to follow up by encoding the categorical data that was present. I did this with label encoding, which suited my data set because a piece of content would often be in more than one genre, for example. So I created new columns for every value, and put a 1 in that column if the content was of that type. I did this for several categories, and found that the vast majority of the content was created in English, and distributed in the United States. I ended up removing this from the dataframe.

I also found which actors appeared frequently on Disney Plus.

No actor appeared more than 30 times, so it may not be a good predictor of the film's rating or score. I will instead be focusing on Genre.

I attempted to use several machine learning techniques to try and make a prediction for the quality of a movie, based on the genres it is a part of, but didn't get very far quite yet. There is not a clear winner for the best technique to use at this time.

# Case Study Part 4

In the previous two sections of this case study, I started by cleaning the data as much as I could, and then tried to create some basic visualizations of the patterns that were present for the content on Disney+. The vast majority of the data that I had available was categorical rather than numeric, so I needed to follow up by encoding the categorical data that was present. I did this with label encoding, which suited my data set because a piece of content would often be in more than one genre, for example. So I created new columns for every value, and put a 1 in that column if the content was of that type. I did this for several categories, and found that the vast majority of the content was created in English, and distributed in the United States. I ended up removing this from the dataframe.

I also found which actors appeared frequently on Disney Plus.

No actor appeared more than 30 times, so it may not be a good predictor of the film's rating or score. I will instead be focusing on Genre.

Last week, I tried to use machine learning to determine if a vector of genres can be indicative of a movie's expected imdb rating. After finding a way to partition the data into a training and testing set, I ran it through a MLPClassifier. I had tried this in the past, but the exercises this week were helpful for getting the data to process properly.

After creating and running the model, I got the loss function down to 2.9. This is strange, and I must have done done something wrong with the neural network. I believe this value is supposed to be less than 1.

I am not conclusively able to tell if if genre is correlated with, or has any relationship with the IMDB rating at this point.