Analyzing the Netflix Dataset

By: Suryaa Rajinikanth

Introduction:

Hey, person judging my SIF application! My name is Suryaa Rajinikanth, and today I'll be outlining some of the findings I came across while examining the Netflix Dataset. I like to watch anime in my free time, so when I saw the dataset about programs on Netflix, I knew that anime in the dataset is something I'd definitely be interested in exploring. Let me give you a list of what I want to accomplish with this dataset.

Section 1:
1. Compare show counts of animes vs similar genres
2. Display and analyze statistics on animes vs shows in general
3. Analyze the addition of animes throughout the years
4. Find the best animes and see what makes them so good

Section 2:
1. Try using NNLM to make a genre prediction

Section 1: Animes in the Dataset

General Anime Stats:

In [2]:
```
1 import pandas as pd
2 import matplotlib.pyplot as plt
```

In [3]:
```
1 file = 'Netflix Dataset Latest 2021.xlsx'
2 netflix = pd.read_excel(file)
3 netflix.shape
```

Out[3]: (9425, 29)

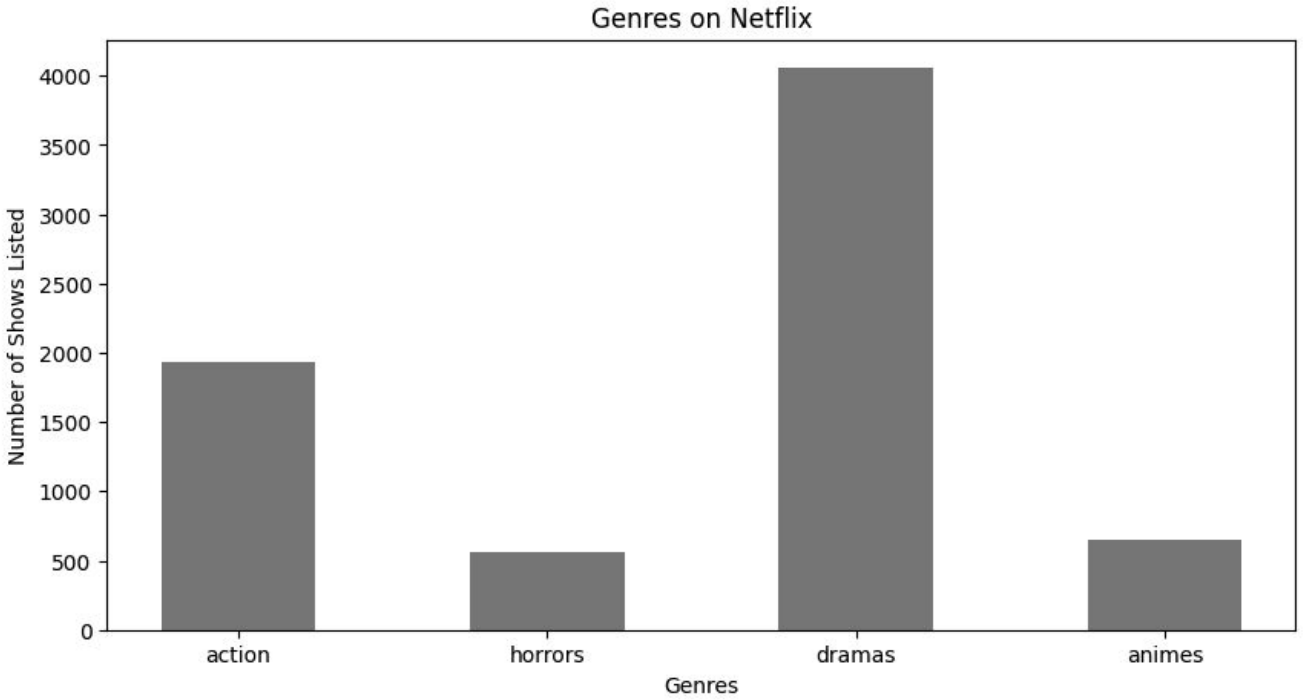
I started by using pandas and matplotlib for parsing the data and effectively displaying my findings. After creating a dataframe, I check its shape and see over 9k shows listed in the dataset- that's a lot!

```
]:  1 animes = netflix.loc[netflix['Tags'].str.contains("anime", case = False, na=False)]
    2 print(str(len(animes.index))+" animes are listed on Netflix.")

651 animes are listed on Netflix.
```

Next, I decided to check how many animes we've got in the dataset. The above code returns 651 by parsing the rows for any tags that contain the word "anime". The percent of animes compared to all shows in this dataset would end up being around 7% which is actually more than I expected.

Let’s Try Some Matplotlib!

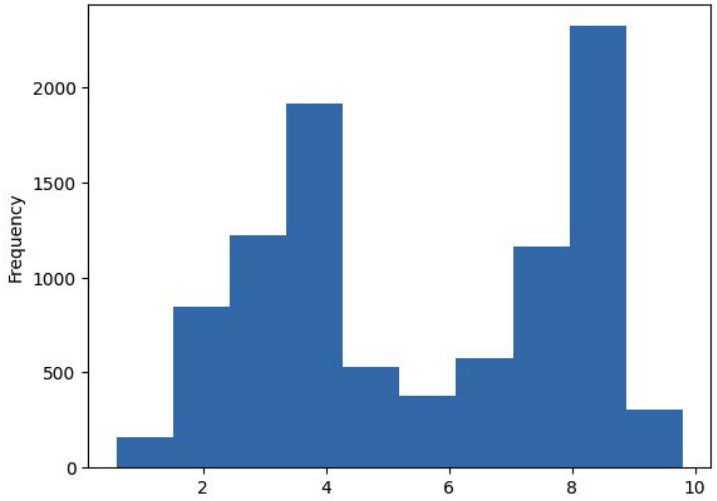


The above graph shows a breakdown of some popular genres compared to animes. I did this by parsing the tags for the name of the genre I wanted to graph and counting the length of that frame. Then I plotted it with matplotlib. I realize that there is a dedicated genres section that I could have searched, but that doesn't consider anime as a genre, which is untrue in my opinion. I think the ratio between genres on this graph is plausible because animes can be dramas or action and they end up being counted in those categories. I did find it surprising that animes outperformed horror movies, but that is plausible given that 7 percent of Netflix titles are animes.

General Netflix Data and Histogram of Gem Scores

Out[6]:

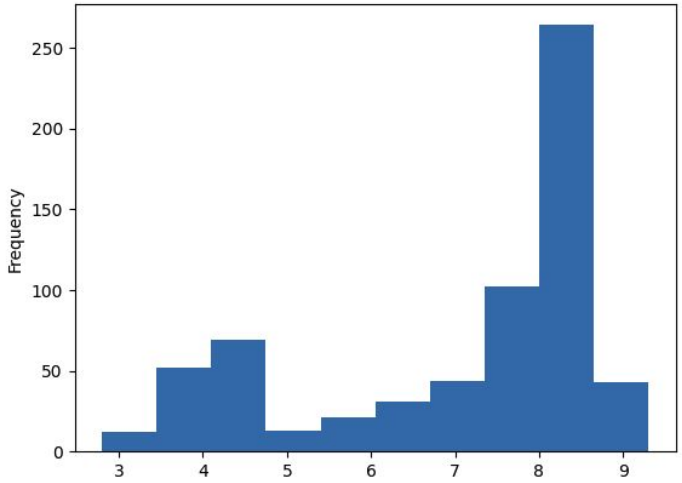
	Hidden Gem Score	IMDb Score	Rotten Tomatoes Score	Metacritic Score	Awards Received	Awards Nominated For	Boxoffice	IMDb Votes
count	9415.000000	9417.000000	5445.000000	4082.000000	5226.000000	6376.000000	3.754000e+03	9.415000e+03
mean	5.540733	6.955517	64.691276	58.113425	9.735936	16.035602	4.845788e+07	6.014725e+04
std	2.447462	0.899681	25.269466	17.143187	19.524116	32.209094	7.243625e+07	1.463837e+05
min	0.600000	1.600000	0.000000	6.000000	1.000000	1.000000	7.200000e+01	5.000000e+00
25%	3.400000	6.500000	49.000000	46.000000	1.250000	2.000000	1.243910e+06	9.695000e+02
50%	5.300000	7.000000	70.000000	59.000000	4.000000	6.000000	2.250466e+07	6.602000e+03
75%	8.100000	7.500000	85.000000	71.000000	9.000000	15.000000	6.425437e+07	5.098700e+04
max	9.800000	9.700000	100.000000	100.000000	300.000000	386.000000	6.593639e+08	2.354197e+06



Anime Specific Data and Histogram of Gem Scores

Out[7]:

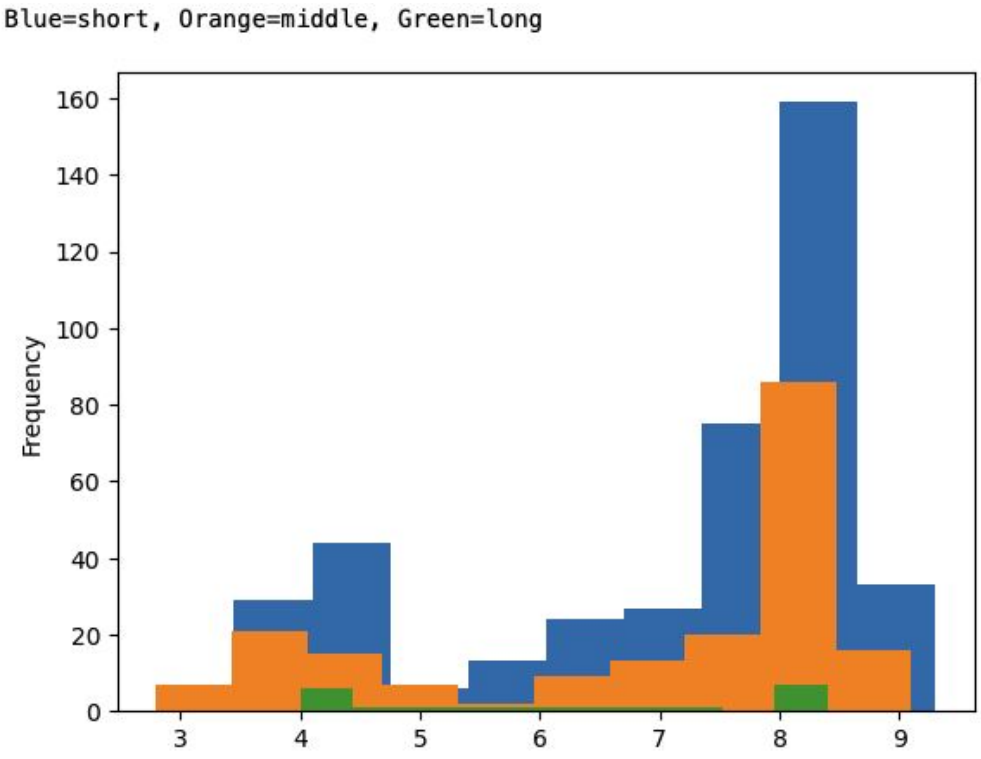
	Hidden Gem Score	IMDb Score	Rotten Tomatoes Score	Metacritic Score	Awards Received	Awards Nominated For	Boxoffice	IMDb Votes
count	651.000000	651.000000	127.000000	57.000000	157.000000	249.000000	8.500000e+01	651.000000
mean	7.039939	7.487404	80.496063	71.719298	3.783439	4.855422	1.689806e+07	14014.827957
std	1.727293	0.609737	12.537483	11.766712	6.318794	8.439728	7.732988e+07	57924.459871
min	2.800000	2.700000	43.000000	43.000000	1.000000	1.000000	6.461000e+03	6.000000
25%	6.000000	7.000000	73.000000	65.000000	1.000000	1.000000	1.721470e+05	500.000000
50%	7.800000	7.500000	82.000000	73.000000	2.000000	2.000000	4.981560e+05	1650.000000
75%	8.300000	7.900000	89.500000	80.000000	3.000000	5.000000	2.250213e+06	5828.000000
max	9.300000	9.100000	100.000000	96.000000	58.000000	69.000000	4.745447e+08	733336.000000



The above histograms and data give an accurate portrayal of what the data is like. Netflix in general tends to be bimodal with peaks at 4 and 8 for GEM scores, giving the total data a mean of 5.54 and a standard deviation of 2.44. Animes are also bimodal to an extent but the peak between 8 and 9 is huge. The mean for this data is 7.03with a standard deviation of 1.72. Animes tend to score higher in the GEM category and are less spread out, evidenced by the higher mean and lower SD. This supports the idea that anime might be one of the better genres in Netflix's catalog.

Section 1.2: Stats Specific to Anime

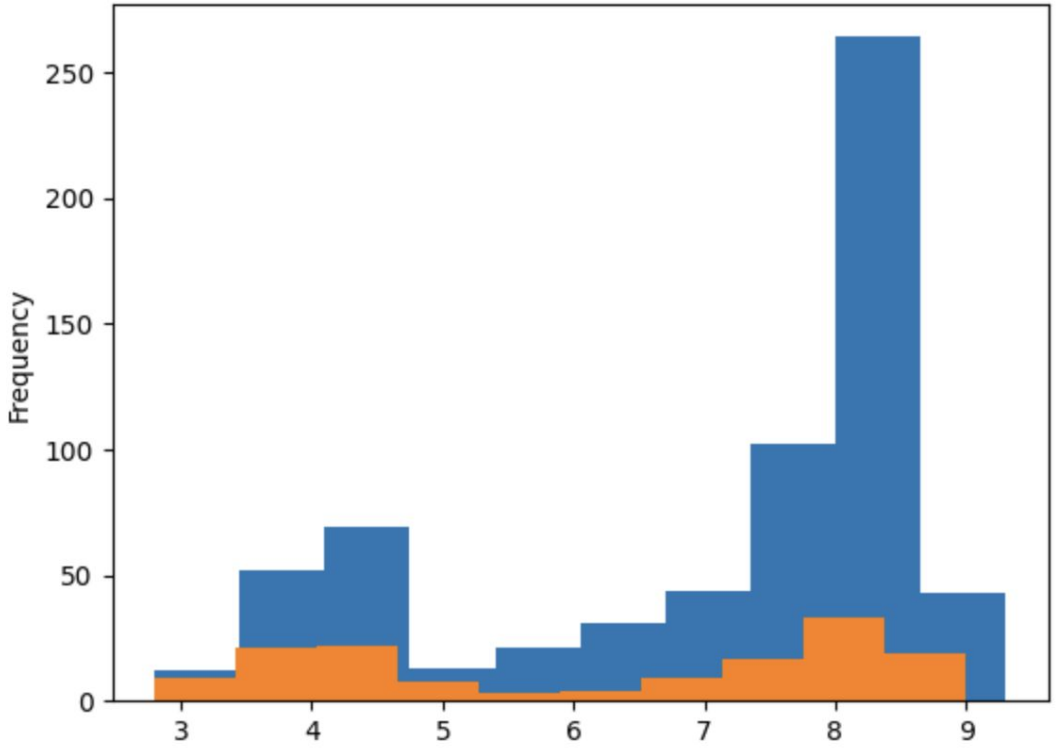
GEM Histogram by Show Runtime



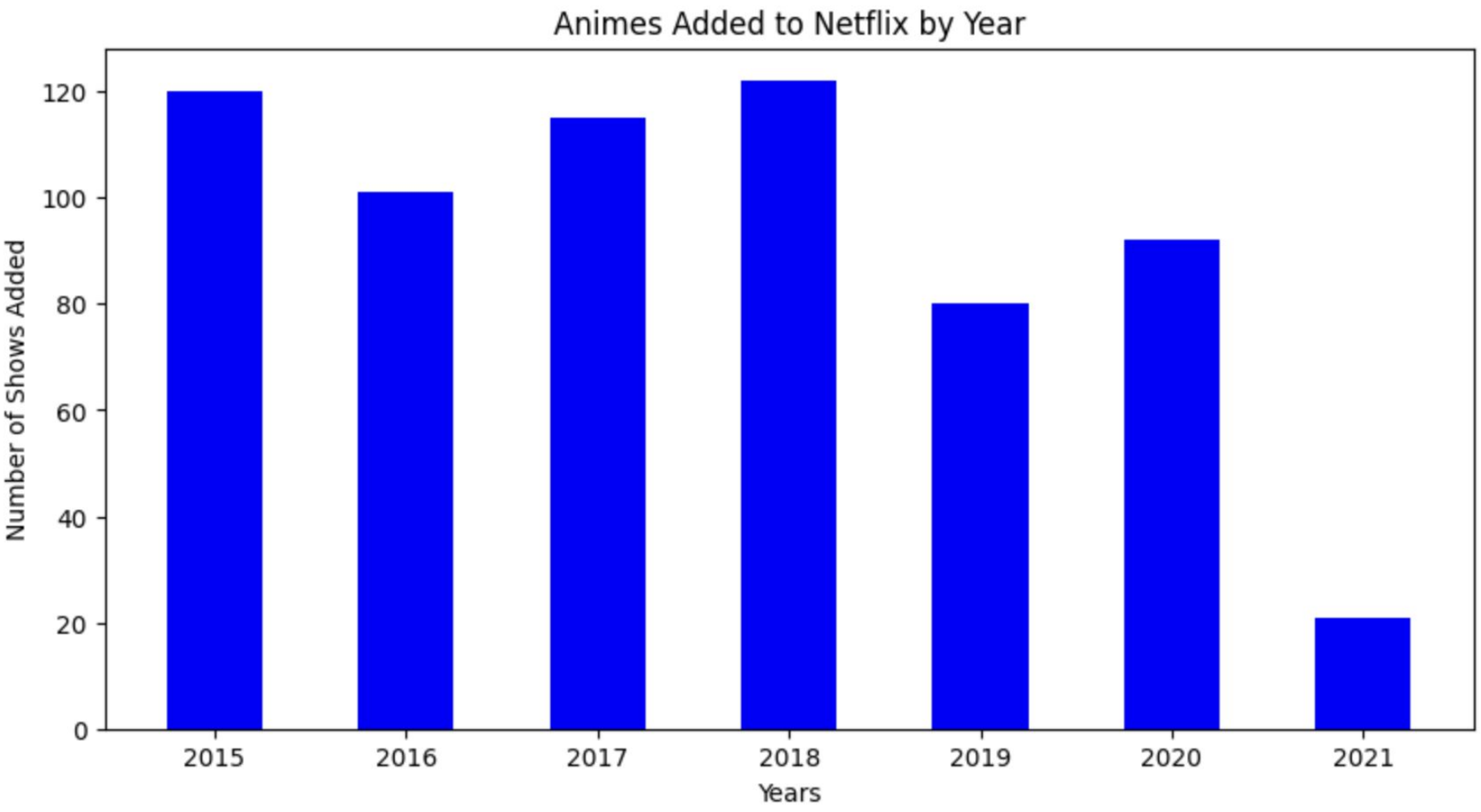
In this histogram, we analyze the spread of shows categorized by length with short being under 30 mins, middle being 1-2 hours, long being over 2 hours. We see that the data is once again bimodal for all groups with peaks around 4 and 8. There do seem to be a lot of short animes, followed by middle length, and finally the long ones with the least amount of data points. Overall, I don’t think that show length really has much of an effect on the GEM score, but the shorter animes tend to have more prominent peaks at the 8 score range.

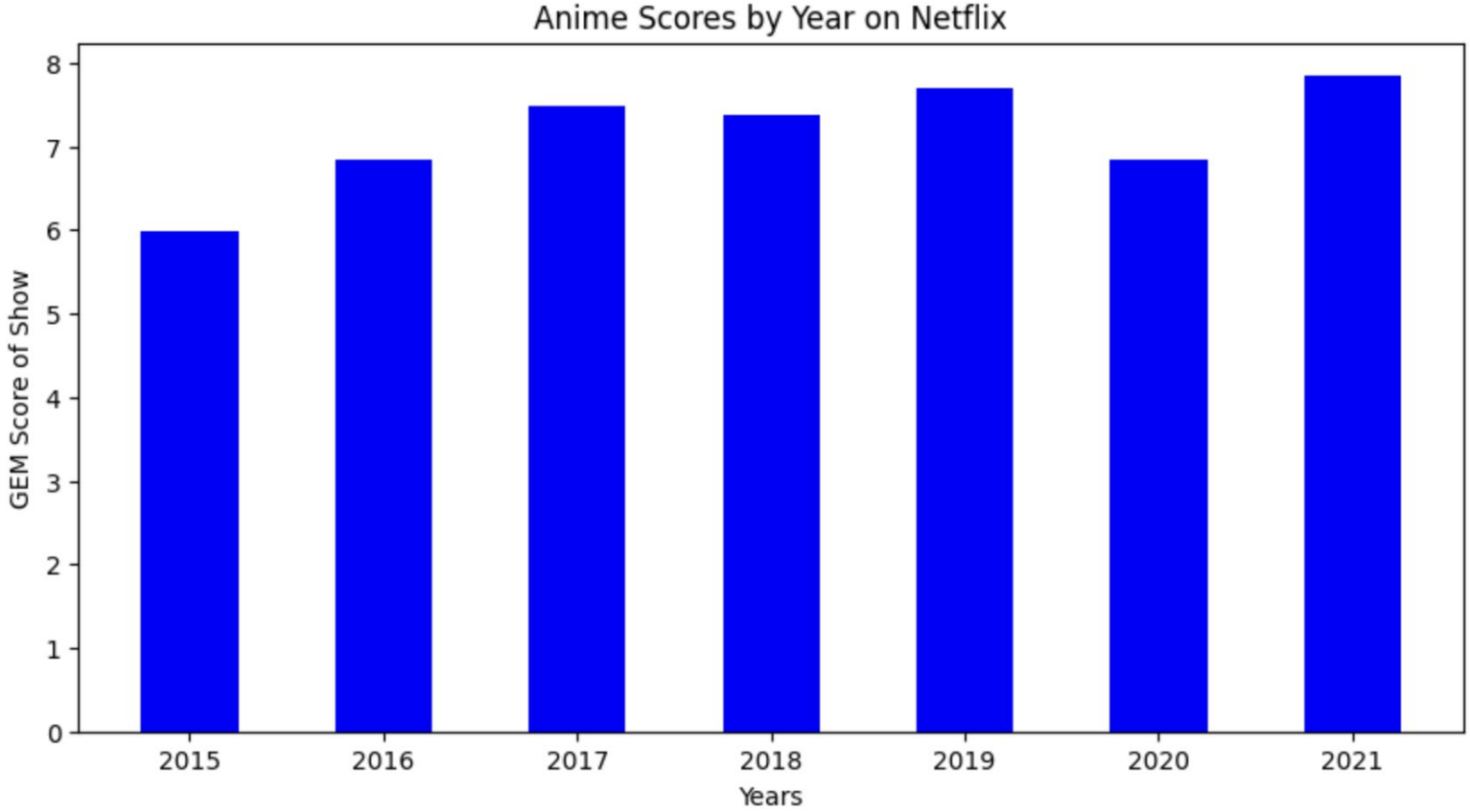
Language vs GEM Score - Orange = Dubbed, Blue = N/A

<AxesSubplot:ylabel='Frequency'>



In this histogram, we attempt to find a trend between whether an anime is dubbed and whether that influences the GEM score. The orange shows animes with english as a language option while blue is only Japanese. The non-dubbed ones end up having a more prominent peak at 8 while the dubbed shows do not really have a peak and are more spread out. This data supports the idea that animes in Japanese tend to have a higher score compared to dubbed animes which are hit or miss and have a high standard deviation and no extremely prominent peaks.


Animes Added to Netflix by Year
Number of Shows Added
0
20
40
60
80
100
120
2015
2016
2017
2018
2019
2020
2021
Years


Anime Scores by Year on Netflix
GEM Score of Show
0
1
2
3
4
5
6
7
8
2015
2016
2017
2018
2019
2020
2021
Years

These two bar graphs show the volume and quality of animes being released by year in the Netflix catalog. In the first bar graph, it is clear that Netflix has slowed down in the release of new animes from 2019 to 2021 going from over 100 to under 100 animes per year in this time. Though this may seem bad on the surface, the second bar graph shows that the quality of the shows through GEM score has gone up since 2015 also, starting at around 6 and trending at 7-8 currently. I believe that the 2021 data has not fully been completed but it is exciting to see that they are releasing animes with the greatest GEM scores to date, nearly reaching an average 8 score.

Summary

In this section, we learned how animes were actually a significant portion of Netflix shows (7%). They are also generally rated better and are less spread out in terms of GEM score compared to all shows. Specifically trying to differentiate the better animes, we found that runtime generally did not have an effect on GEM score, but there were many more shorter shows than longer. The language did have some effect on GEM score, discernable on a histogram. Finally, we observed that Netflix was slowing down a bit on releasing animes, but the ratings have gotten progressively better since 2015.

Section 2: Using NNLM to Make a Genre Prediction

```
netflix1 = netflix[["Genre", "Summary"]]
genres = list()
for index, row in netflix1.iterrows():
  inner_genres = (str(row['Genre'])).split(", ")
  for inner_genre in inner_genres:
    genres.append(inner_genre)
true_genres = np.unique(genres)

len(true_genres)

```

```
29
```

To perform this task, I decided to use Tensorflow and make a prediction of the genre using the summary. I found there were 29 unique genres using this code to the left.

nture	Animation	Biography	Comedy	Crime	Documentary	Drama	Family	...	Romance	Sci-Fi	Short	Sport	Talk-Show	Thriller	War	Western	nan	Summary
0	0	0	0	1	0	1	0	...	1	0	0	0	0	0	0	0	0	A med student with a supernatural gift tries t...
0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	When nerdy Johanna moves to London, things get...
0	0	0	1	0	0	0	0	...	1	0	0	0	0	0	0	0	0	After her ex-boyfriend cons her out of a large...

Next, I created a new dataframe and marked which genres were contained by each show along with the summary on the right. The above is the result of a dataframe.head(3).

Creating a Model

```
embedding_layer = hub.KerasLayer("https://tfhub.dev/google/tf2-preview/nnlm-en-dim128/1"
embedding_layer(netflix2.iloc[test_row,:]["Summary"][:1].to_numpy())

BERT = tf.keras.Sequential()
BERT.add(embedding_layer)
BERT.add(tf.keras.layers.Dense(64, activation='relu'))
BERT.add(tf.keras.layers.Dense(len(true_genres), activation='sigmoid'))
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 128)               124642688

 dense (Dense)               (None, 64)                8256

 dense_1 (Dense)             (None, 29)                1885

=================================================================
Total params: 124,652,829
Trainable params: 10,141
Non-trainable params: 124,642,688
_________________________________________________________________
```

I defined my model by creating a train tensor and test tensor. In this case, I ended up using 5 percent of the rows for testing and the other 95 percent for training. I also added a KerasLayer using TensorFlow Hub's nnlm-en-dim128 text embedding module. Training with this will provide text analysis on the summary, which we need.

```
Epoch 1/25
1/1 [==============================] - 10s 10s/step - loss: 0.6000 - accuracy: 0.0000e+00 - precision_1: 0.1951 - r
ecall_1: 0.4000 - val_loss: 0.5986 - val_accuracy: 0.0000e+00 - val_precision_1: 0.2119 - val_recall_1: 0.4167
```

```
Epoch 25/25
1/1 [==============================] - 9s 9s/step - loss: 0.5551 - accuracy: 0.0000e+00 - precision_1: 0.3953 - rec
all_1: 0.2833 - val_loss: 0.5527 - val_accuracy: 0.0000e+00 - val_precision_1: 0.4250 - val_recall_1: 0.2833
```

Now, we train the model on our dataset- in this case, I chose 25 epochs. In the first epoch we see an accuracy of just 21.2%. However, by the end, we see a massive improvement to 42.5% accuracy for prediction of genres based on just analyzing the text in the summary. Though this figure may seem rather low, I believe that with the training done on this limited dataset, it is rather impressive that we reached an accuracy this high. I will discuss further improvements, but there does seem to be a trend between genre and the summary. We can break down which genres are easiest to predict too.

	precision	recall	f1-score	support
Action	0.00	0.00	0.00	1
Adult	0.00	0.00	0.00	0
Adventure	0.00	0.00	0.00	5
Animation	0.00	0.00	0.00	3
Biography	0.00	0.00	0.00	0
Comedy	1.00	0.38	0.55	8
Crime	0.00	0.00	0.00	2
Documentary	0.00	0.00	0.00	1
Drama	0.89	0.67	0.76	12
Family	1.00	0.43	0.60	7
Fantasy	0.00	0.00	0.00	7
Film-Noir	0.00	0.00	0.00	0
Game-Show	0.00	0.00	0.00	0
History	1.00	1.00	1.00	1
Horror	0.00	0.00	0.00	1
Music	0.00	0.00	0.00	0
Musical	0.00	0.00	0.00	1
Mystery	0.00	0.00	0.00	0
News	0.00	0.00	0.00	0
Reality-TV	0.00	0.00	0.00	0
Romance	1.00	0.25	0.40	8
Sci-Fi	0.00	0.00	0.00	0
Short	0.00	0.00	0.00	1
Sport	0.00	0.00	0.00	0
Talk-Show	0.00	0.00	0.00	0
Thriller	0.00	0.00	0.00	1
War	0.00	0.00	0.00	1
Western	0.00	0.00	0.00	0
nan	0.00	0.00	0.00	0
micro avg	0.42	0.28	0.34	60
macro avg	0.17	0.09	0.11	60
weighted avg	0.58	0.28	0.37	60
samples avg	0.37	0.29	0.28	60

Wow- there seems to be a great rift in the genres this model can and cannot predict. Comedy, Drama, Family, History, and Romance all have a 100% or near perfect accuracy. This makes sense, as a summary would likely include keywords that hint to their respective genre. It is not implausible to say that there is indeed a trend between some genres and their summaries because we were able to make such accurate predictions using this data.

Summary

Using nnlm and training this model to discover a trend between genre and text in the summary yielded a result that there is indeed a relationship between these two characteristics of a Netflix show. However, in the future I would like to use a larger dataset to train the model and possibly factor in GEM score so that the accuracy can go up and shows can be differentiated easily. Additionally, I would like my training data and test data to be more representative of each genre. I ended up just choosing 5 percent of the rows to be my test data, but this could have only represented some genres and not others. This may be why some genres had 0% accuracy and others were really high- the more common genres may have been favored in the tests. Overall, the test was still promising though.