Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

J Component report

Programme : **B.Tech (CSE: CORE)**

Course Title : **Foundation of Data Analytics**

Course Code : **CSE3505**

Slot : **F2**

Title: Streaming Content Dashboard

Team Members: **KASHISH BAJAJ (20BCE1790)**

AKSHIT JAIN (20BCE1818)

AKASH RAJ BEHERA (20BCE1829)

Faculty: PRIYADARSHINI R **Sign:**

Date:

Streaming Content Dashboard

20BCE1790, KASHISH BAJAJ, Vellore Institute of Technology, Chennai, India

20BCE1818, AKSHIT JAIN, Vellore Institute of Technology, Chennai, India

20BCE1829, AKASH RAJ BEHERA, Vellore Institute of Technology, Chennai, India

ABSTRACT

As we all know in today's world data analysis and visualization is becoming important thing because of the way the human brain processes information, using charts or graphs to visualize large amounts of complex data is easier than poring over spreadsheets or reports. Data visualization is a quick, easy way to convey concepts in a universal manner – and you can experiment with different scenarios by making slight adjustments. Now a days people do not want to waste any time on viewing bad shows and they first look at the ratings and later they decide what to see. According to this situation we designed our project to make streaming content dashboard which will enable us to visualize all the famous shows in every aspect we can understand in a clear way. We also clustered the combined data from Netflix, Hulu, Disney Plus and Amazon Prime using K-Means and created a recommendation system to find similar movies to what the viewer has watched.

KEYWORD

Netflix, Hulu, Disney Plus, Amazon Prime, recommendation system, text clustering, data visualization and analytics, OTT Content and k-means algorithms

1. INTRODUCTION

Recommender Systems (RSs) are characterized by the capability of filtering large information spaces and selecting the items that are likely to be more interesting and attractive to a user.

OTT Platforms are the biggest users of recommendation systems. So, in this Project we aim to visualize the content library of top OTT Platforms like Netflix, Disney Plus, Hulu and Amazon Prime. While doing this we will also discover correlations and recurring patterns in the dataset with interesting inferences.

Finally, we will see how the recommendation engine works to deliver similar content as quickly as possible.

2. About The Dataset

For this project we will use 4 datasets containing of listings of all the movies and tv shows available on Netflix, Hulu, Disney Plus and Amazon Prime, along with details such as - cast, directors, ratings, release year, duration, etc. In total there are approximately 22k observations. It is obtained from Kaggle Open-Source Dataset Library (Source).

2.1 Feature components for analysis & visualization

For this visualization and analysis, we use feature attributes from the dataset, namely,

- Type
- Title
- Director
- Cast
- Country
- Date Added
- Release Year
- Rating
- Duration
- Listed In
- Description

Each individual dataset contains all the following attributes. During the project we will combine all 4 datasets into one and then we will append a column denoting the OTT platform.

3. Design and flow of models

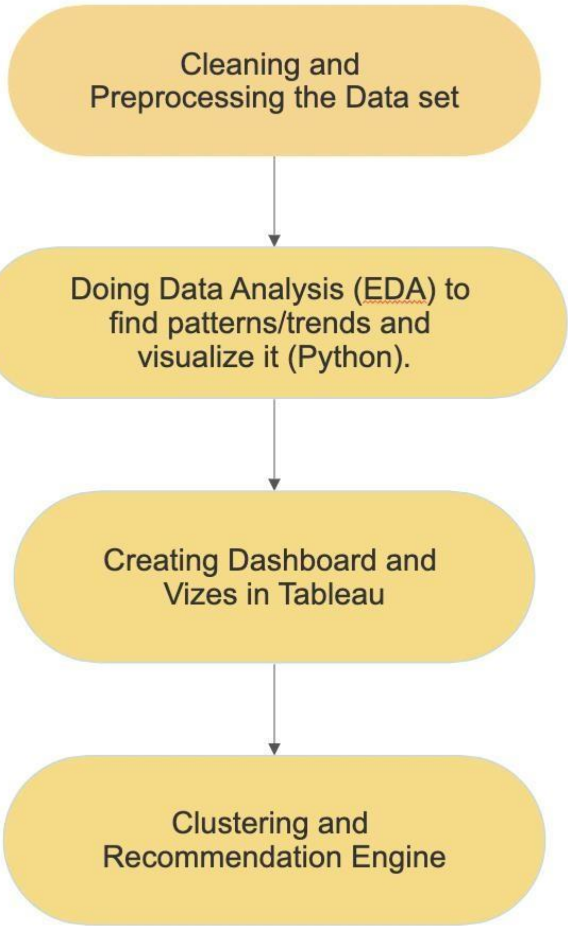


Fig.1 design and flow of model

For the Visualization we have used the following modules and analysis parameters:

3.1 Module 1: data cleaning and dataset analysis

After Importing the data set, we need to clean it and analyze what data we were able to collect. After this we can easily plan which parameters to visualize.

3.2 Module 2: Doing Data Analysis (EDA) to find patterns/trends and visualize it (Python).

The attributes from the obtained data set are compared with each other to find correlations and dependencies and then these are visualized using different types of graphs. We can use these graphs to visualize common trends in the dataset.

3.3 Module 3: Creating Dashboard and Vizes in Tableau

We then use Tableau to further Visualize the Dataset and create interactive Dashboards. We found Tableau to be an incredibly versatile and powerful tool for this purpose.

3.4 Module 4: Clustering and Recommendation Engine

We will use K-Means clustering to cluster similar data. We then append the cluster id generated to the combined dataset to facilitate the recommendation engine

K-means

K-means algorithm is an iterative algorithm that tries to partition the dataset into *K* pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.

The way kmeans algorithm works is as follows:

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids.
4. Compute the sum of the squared distance between data points and all centroids.
5. Assign each data point to the closest cluster (centroid).

Compute the centroids for the clusters by taking the average of the all-data points that belong to each cluster.
The objective function is:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2$$

Recommendation Engine

The recommendation Engine takes a Movie or Show Title as an input. It then finds the cluster id of that entry. It uses the cluster id to reduce the search space.

Now it runs a text similarity check between the description of entered show or movie to find similar content from that cluster.

Thus, using K-Means and Text Similarity, it achieves fast and accurate results.

4. IMPLEMENTATION

4.1 First we import modules and datasets.

Importing all the libraries and modules

First import the libraries to better analyze the data set. Here matplotlib and plotly are used for visualization and word cloud.

```
[ ] import pandas as pd
    import numpy as np
    import plotly.express as px
    import plotly.graph_objects as go
    import matplotlib.pyplot as plt
    import seaborn as sns
    import plotly.io as pio
    from plotly.offline import iplot
    from plotly.subplots import make_subplots
    from wordcloud import WordCloud, STOPWORDS
    import random
    import re
```

Add text cell

In the following codes

df1- *Amazon Prime Dataset*

df2- *Hulu Dataset*

df3 - *Disney Plus Dataset*

df4 - *Netflix Dataset*

+ Code + Text

```
[ ]
    df1 = pd.read_csv("amazon_prime_titles.csv", delimiter=",", encoding="latin-1", parse_dates=["date_added"], index_col=["show_id"])
    df2 = pd.read_csv("hulu_titles.csv", delimiter=",", encoding="latin-1", parse_dates=["date_added"], index_col=["show_id"])
    df3 = pd.read_csv("disney_plus_titles.csv", delimiter=",", encoding="latin-1", parse_dates=["date_added"], index_col=["show_id"])
    df4 = pd.read_csv("netflix_titles.csv", delimiter=",", encoding="latin-1", parse_dates=["date_added"], index_col=["show_id"])
```

4.2 Dataset Analysis

```
df1.dtypes

type                    object
title                   object
director                object
cast                    object
country                 object
date_added      datetime64[ns]
release_year             int64
rating                  object
duration                object
listed_in               object
description             object
dtype: object
```

```
[ ] df2.dtypes

type                    object
title                   object
director                object
cast                   float64
country                 object
date_added      datetime64[ns]
release_year             int64
rating                  object
duration                object
listed_in               object
description             object
dtype: object
```

```
[ ] df3.dtypes

type                    object
title                   object
director                object
cast                    object
country                 object
date_added      datetime64[ns]
release_year             int64
rating                  object
duration                object
listed_in               object
description             object
dtype: object
```

```
[ ] print("The size and shape of dataset 1")
    print(df1.size)
    print(df1.shape)

The size and shape of dataset 1
106348
(9668, 11)
```

```
[ ] print("The size and shape of dataset 2")
    print(df2.size)
    print(df2.shape)

The size and shape of dataset 2
33803
(3073, 11)
```

```
[ ] print("The size and shape of dataset 3")
    print(df3.size)
    print(df3.shape)

The size and shape of dataset 3
15950
(1450, 11)
```

```
[ ] print("The size and shape of dataset 4")
    print(df4.size)
    print(df4.shape)

The size and shape of dataset 4
96877
(8807, 11)
```

```
df4.dtypes

type                    object
title                   object
director                object
cast                    object
country                 object
date_added      datetime64[ns]
release_year             int64
rating                  object
duration                object
listed_in               object
description             object
dtype: object
```

Here we can see the size and attributes of the dataset. All data is in correct form except cast in df2 (Hulu) which is in float64 format. We will resolve that in pre processing

4.3 Data Cleaning and Preprocessing

DATA CLEANING

We will go through all 4 datasets to clean them.

```
df1["date_added"] = df1["date_added"].dt.year
df1["date_added"].unique()
```

```
array([2021.,   nan])
```

```
df1["date_added"].fillna(0, inplace=True)
df1["date_added"] = df1["date_added"].astype(int)
df1.head()
```

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
s1	Movie	The Grand Seduction	Don McKellar	Brendan Gleeson, Taylor Kitsch, Gordon Pinsent	Canada	2021	2014	NaN	113 min	Comedy, Drama	A small fishing village must procure a local d...
s2	Movie	Take Care Good Night	Girish Joshi	Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar	India	2021	2018	13+	110 min	Drama, International	A Metro Family decides to fight a Cyber Crimin...
s3	Movie	Secrets of Deception	Josh Webber	Tom Sizemore, Lorenzo Lamas, Robert LaSardo, R...	United States	2021	2017	NaN	74 min	Action, Drama, Suspense	After a man discovers his wife is cheating on ...
s4	Movie	Pink: Staying True	Sonia Anderson	Interviews with: Pink, Adele, BeyoncÃ©, Britne...	United States	2021	2014	NaN	69 min	Documentary	Pink breaks the mold once again, bringing her ...
s5	Movie	Monster Maker	Giles Foster	Harry Dean Stanton, Kieran O'Brien, George Cos...	United Kingdom	2021	1989	NaN	45 min	Drama, Fantasy	Teenage Matt Banting wants to work with a famo...

```
df1.loc[df1["date_added"]==0, ]
```

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
s17	Movie	Zoombies	Glenn Miller	Marcus Anderson, Kaiwi Lyman, Andrew Asper	NaN	0	2016	13+	87 min	Horror, Science Fiction	When a strange virus quickly spreads through a...
s18	TV Show	Zoo Babies	NaN	Narrator - Gillian Barlett	NaN	0	2008	ALL	1 Season	Kids, Special Interest	A heart warming and inspiring series that welc...
s19	TV Show	ZoÃ« Coombs Marr: Bossy Bottom	NaN	ZoÃ« Coombs Marr	NaN	0	2020	18+	1 Season	Comedy, Talk Show and Variety	ZoÃ« Coombs Marr has been on hiatus. Sort of. ...

Here we transform the date_added and date_released fields to extract years from it. We also check for and remove Null values. We do the same for all dataset.

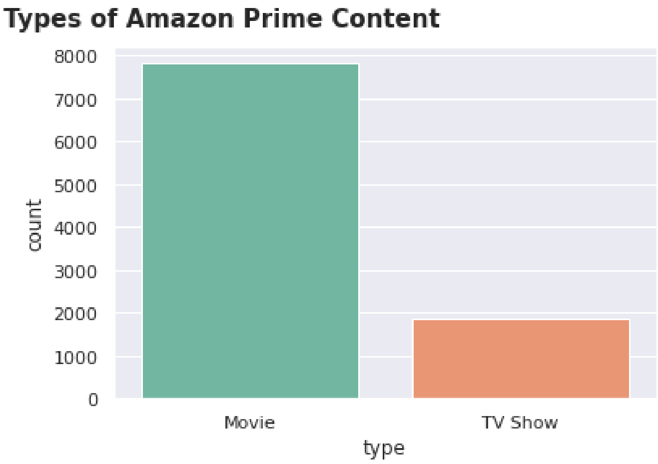
```
[ ] df1.info()

    <class 'pandas.core.frame.DataFrame'>
    Index: 9668 entries, s1 to s9668
    Data columns (total 11 columns):
     #   Column        Non-Null Count  Dtype
    ---  ------        --------------  -----
     0   type          9668 non-null   object
     1   title         9668 non-null   object
     2   director      7586 non-null   object
     3   cast          8435 non-null   object
     4   country       672 non-null    object
     5   date_added    9668 non-null   int64
     6   release_year  9668 non-null   int64
     7   rating        9331 non-null   object
     8   duration      9668 non-null   object
     9   listed_in     9668 non-null   object
     10  description   9668 non-null   object
    dtypes: int64(2), object(9)
    memory usage: 906.4+ KB

[ ] df1.duplicated().sum()

    0

[ ] df1.fillna("No Data", inplace=True)
    df1.isnull().sum()

    type            0
    title           0
    director        0
    cast            0
    country         0
    date_added      0
    release_year    0
    rating          0
    duration        0
    listed_in       0
    description     0
    dtype: int64
```

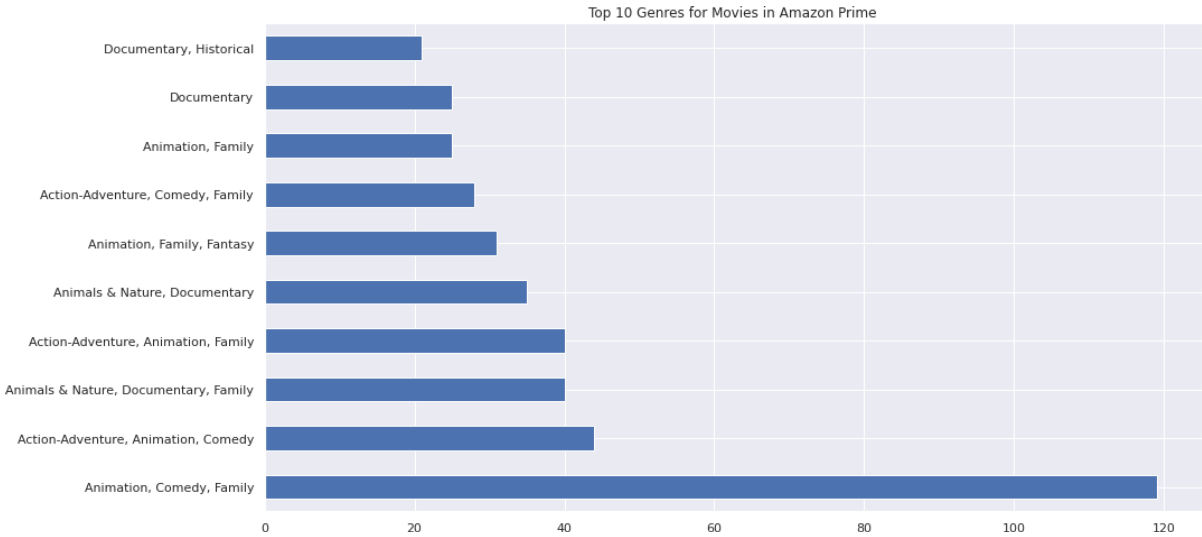
FOR DATASET 2(HULU), we will also convert float64 to string

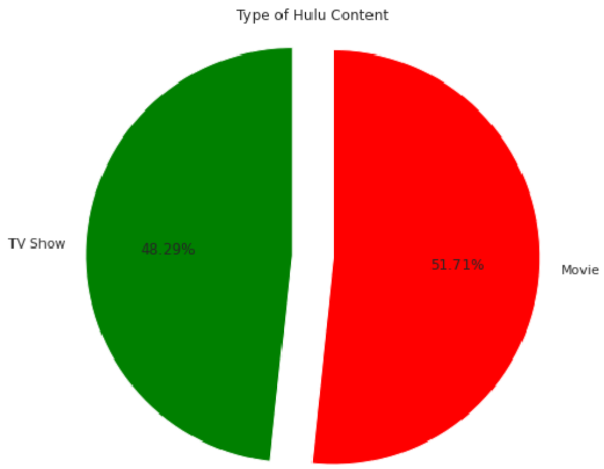
```
[ ] df2['cast'] = df2['cast'].astype(str)
```

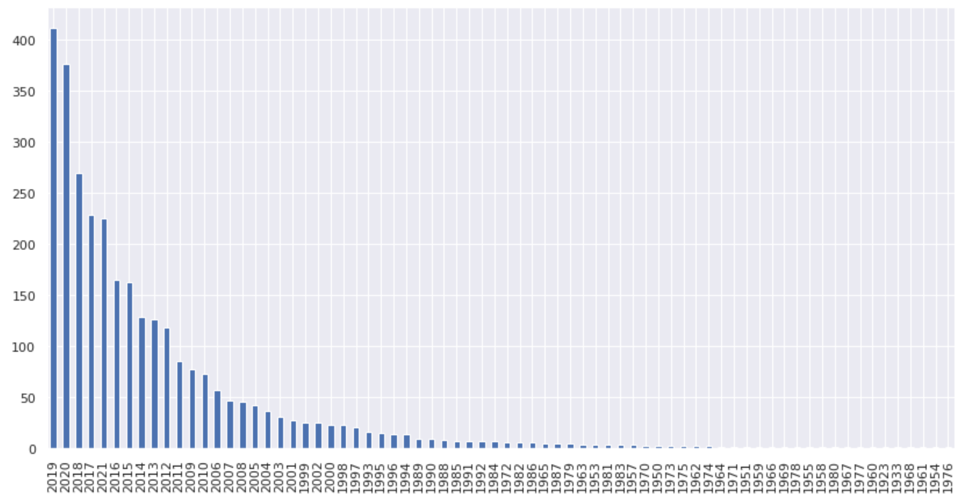
Here, we check the datasets for null and duplicated values as well as missing data. We also convert the float64 column from Hulu dataset to String format.

4.4 Visualization of Datasets

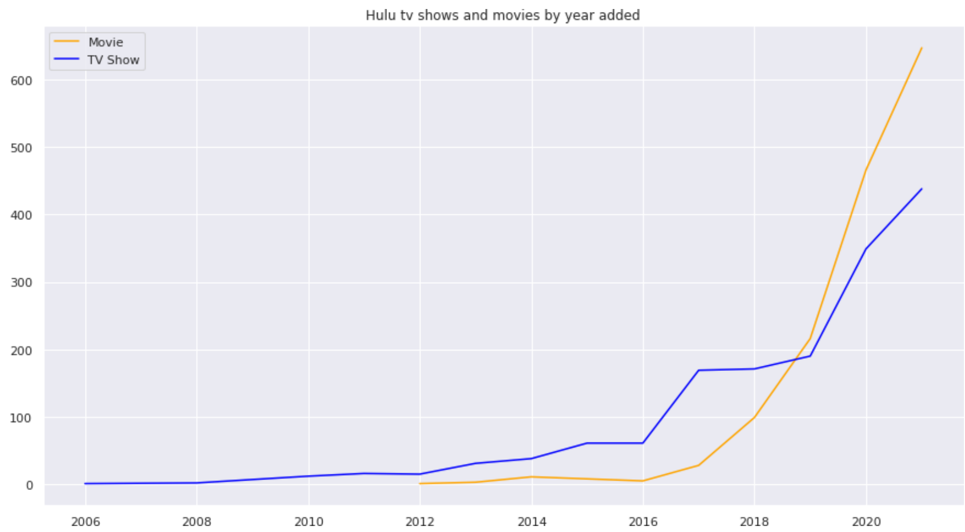

Types of Amazon Prime Content
8000
7000
6000
5000
4000
3000
2000
1000
0
count
Movie
TV Show
type

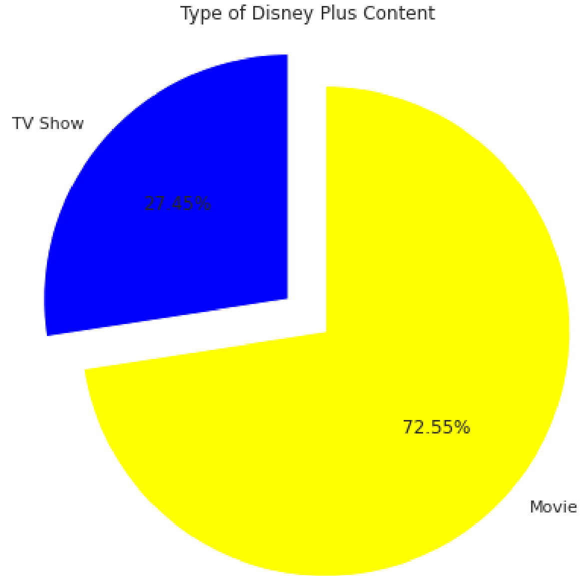

Top 10 Genres for Movies in Amazon Prime
Documentary, Historical
Documentary
Animation, Family
Action-Adventure, Comedy, Family
Animation, Family, Fantasy
Animals & Nature, Documentary
Action-Adventure, Animation, Family
Animals & Nature, Documentary, Family
Action-Adventure, Animation, Comedy
Animation, Comedy, Family
0
20
40
60
80
100
120

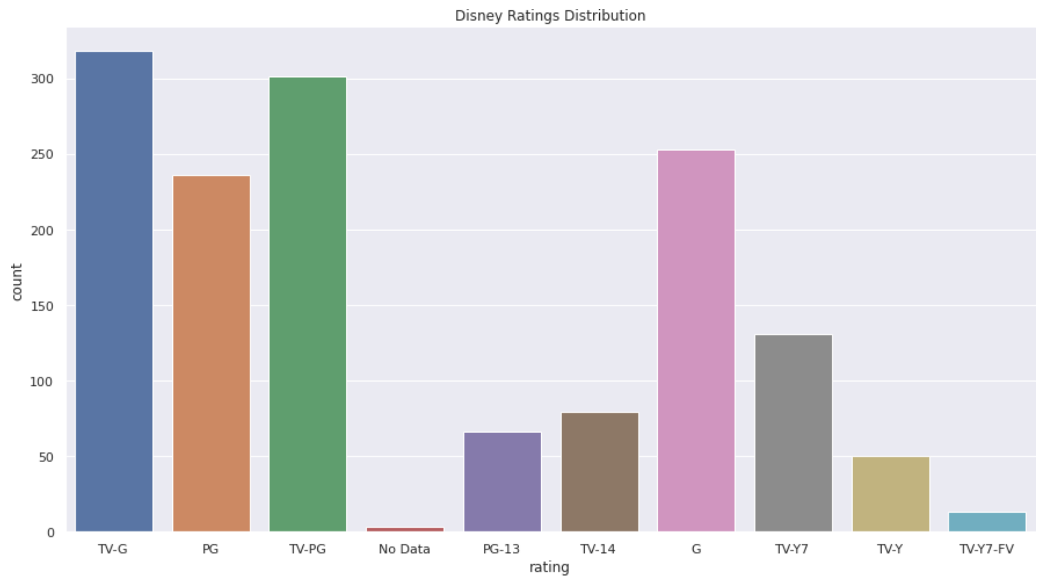


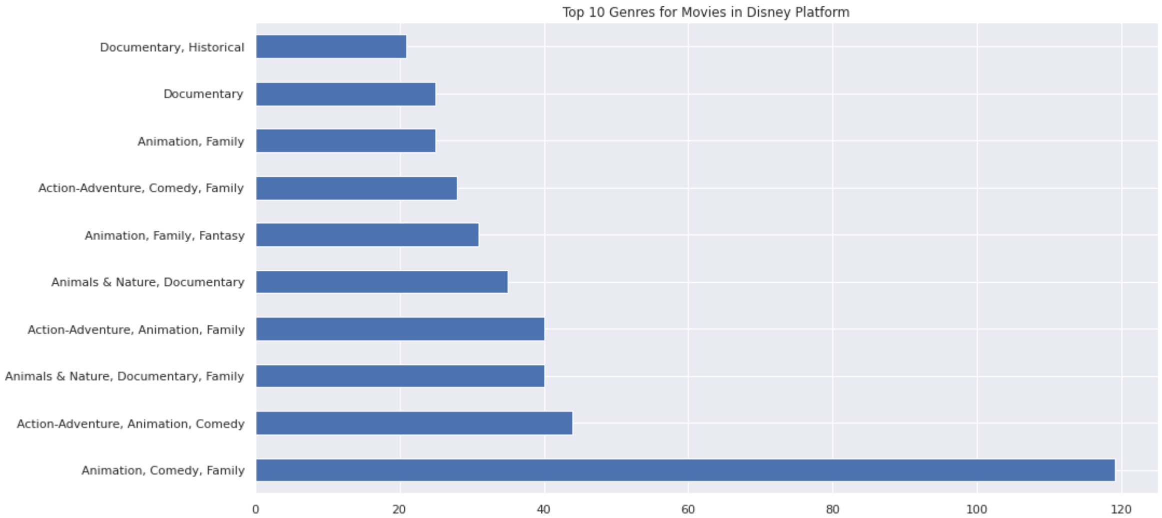


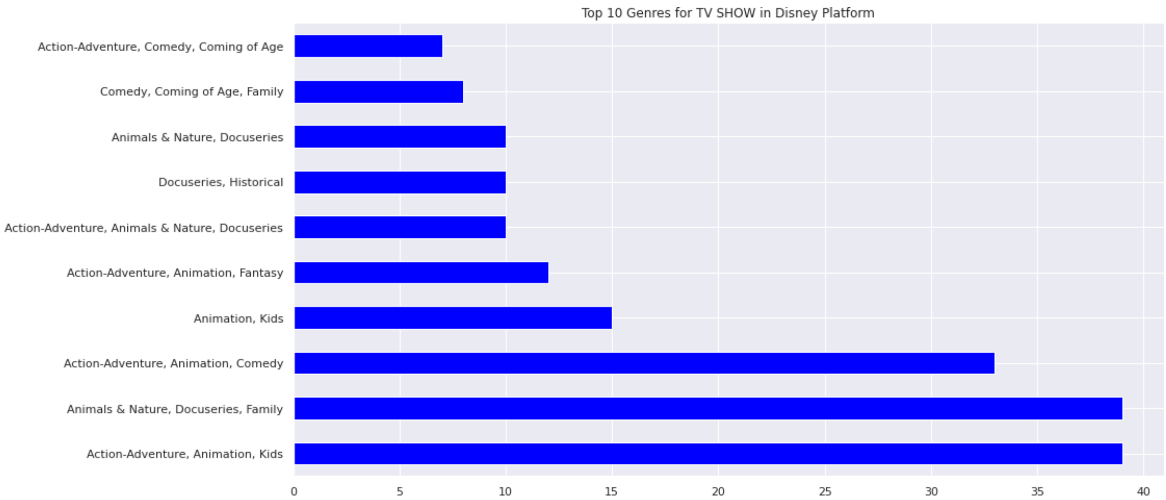
Content by their Release year on Hulu

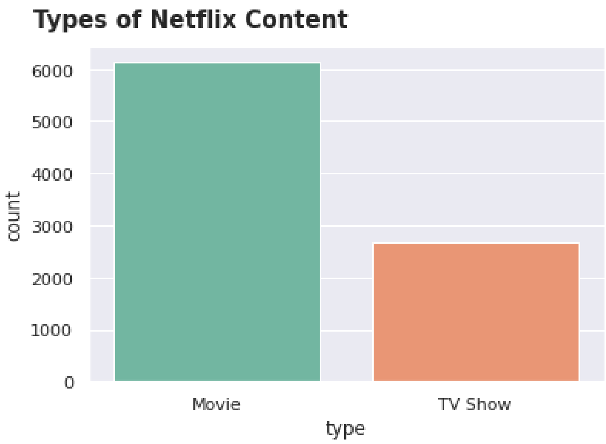

Hulu tv shows and movies by year added
Movie
TV Show
600
500
400
300
200
100
0
2006
2008
2010
2012
2014
2016
2018
2020

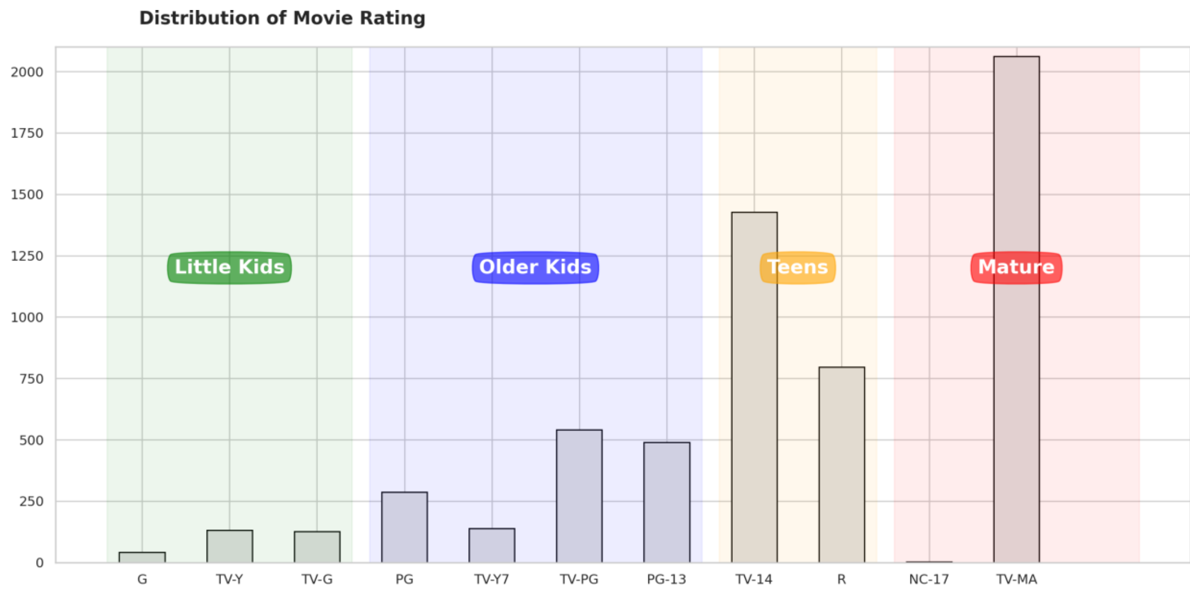

Type of Disney Plus Content
TV Show
27.45%
72.55%
Movie

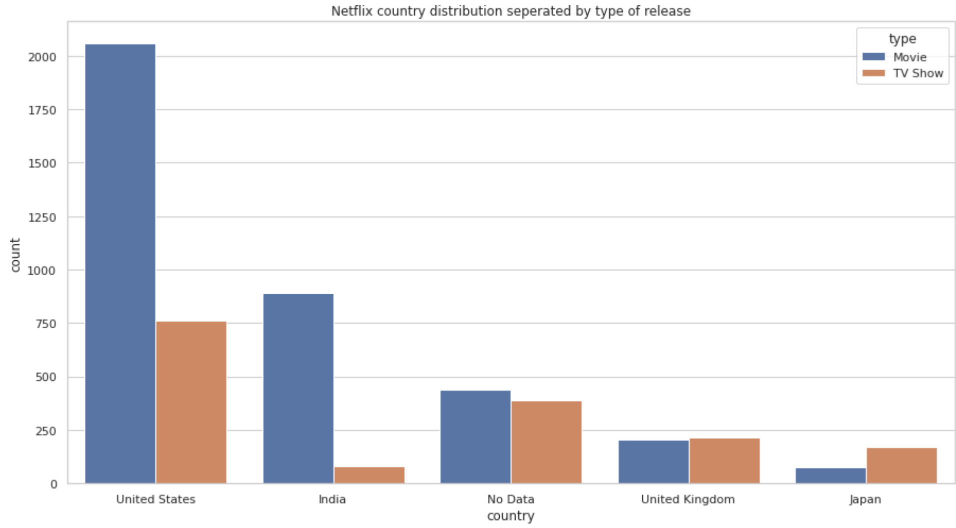

Disney Ratings Distribution
count
300
250
200
150
100
50
0
TV-G
PG
TV-PG
No Data
PG-13
TV-14
G
TV-Y7
TV-Y
TV-Y7-FV
rating

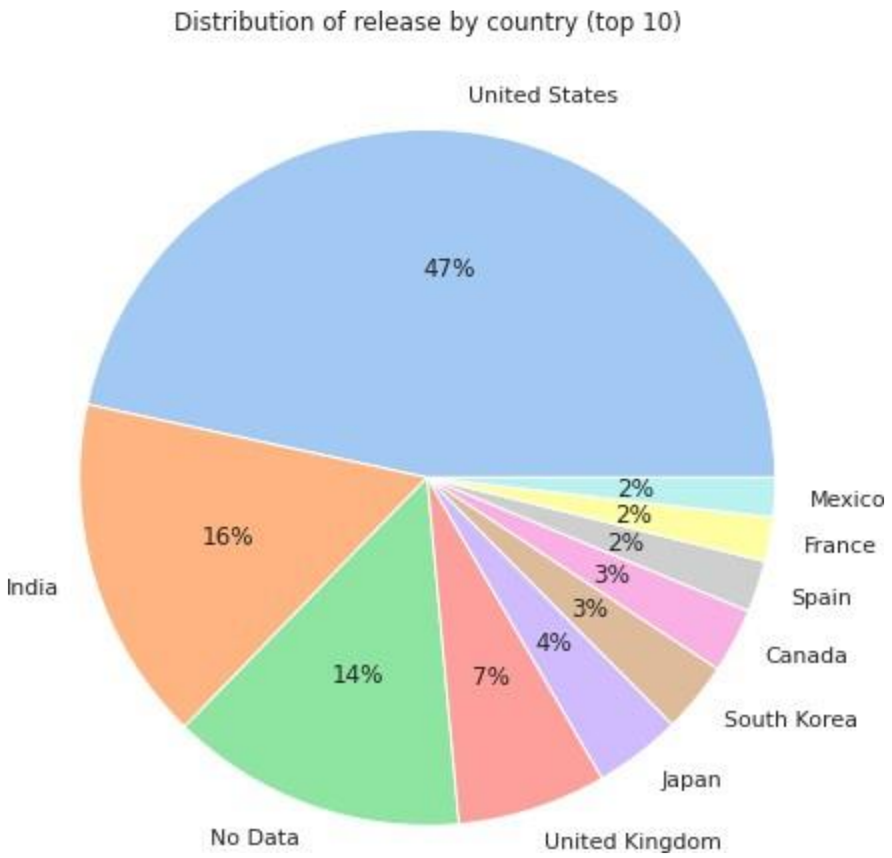

Top 10 Genres for Movies in Disney Platform
Documentary, Historical
Documentary
Animation, Family
Action-Adventure, Comedy, Family
Animation, Family, Fantasy
Animals & Nature, Documentary
Action-Adventure, Animation, Family
Animals & Nature, Documentary, Family
Action-Adventure, Animation, Comedy
Animation, Comedy, Family
0
20
40
60
80
100
120


Top 10 Genres for TV SHOW in Disney Platform
Action-Adventure, Comedy, Coming of Age
Comedy, Coming of Age, Family
Animals & Nature, Docuseries
Docuseries, Historical
Action-Adventure, Animals & Nature, Docuseries
Action-Adventure, Animation, Fantasy
Animation, Kids
Action-Adventure, Animation, Comedy
Animals & Nature, Docuseries, Family
Action-Adventure, Animation, Kids
0
5
10
15
20
25
30
35
40

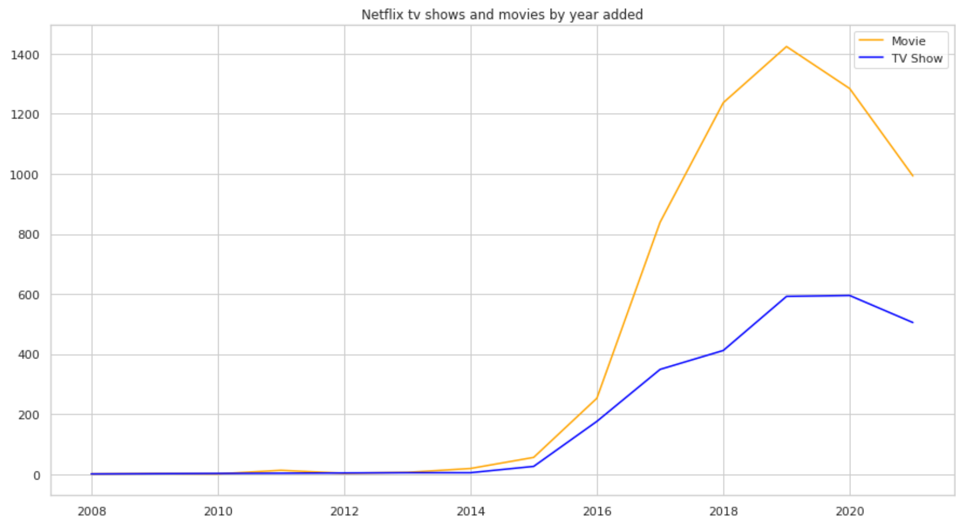

Types of Netflix Content
count
6000
5000
4000
3000
2000
1000
0
Movie
TV Show
type


Distribution of Movie Rating
2000
1750
1500
1250
1000
750
500
250
0
Little Kids
Older Kids
Teens
Mature
G
TV-Y
TV-G
PG
TV-Y7
TV-PG
PG-13
TV-14
R
NC-17
TV-MA

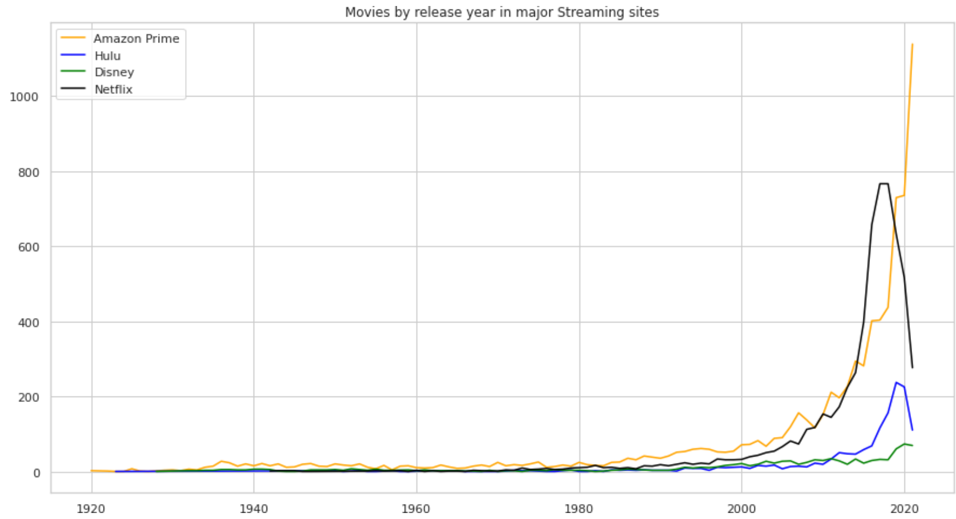

Netflix country distribution seperated by type of release
type
Movie
TV Show
count
2000
1750
1500
1250
1000
750
500
250
0
United States
India
No Data
United Kingdom
Japan
country

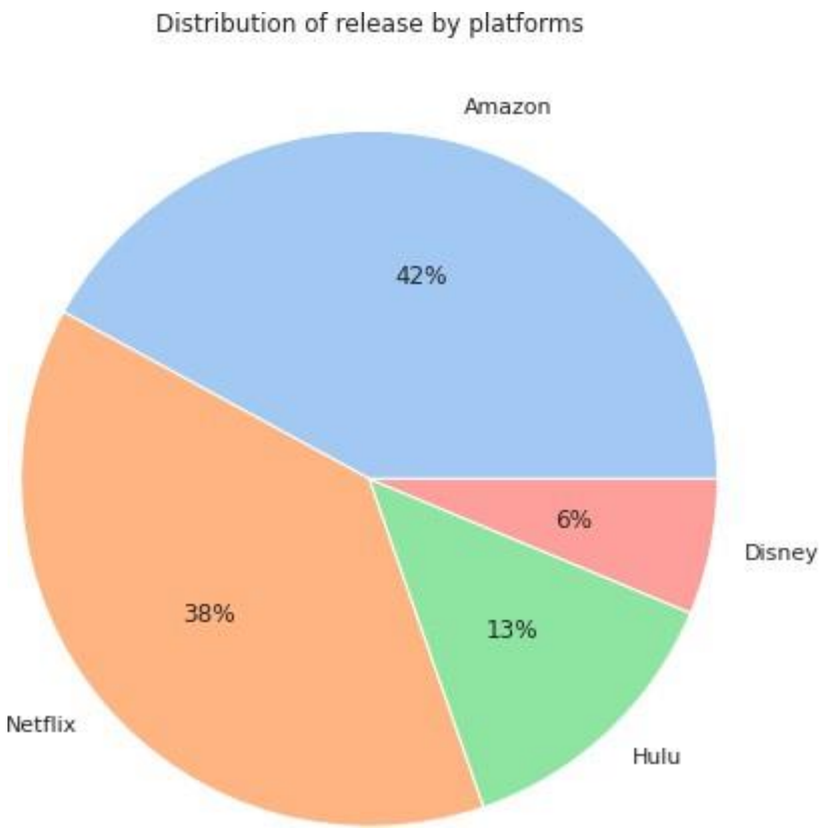


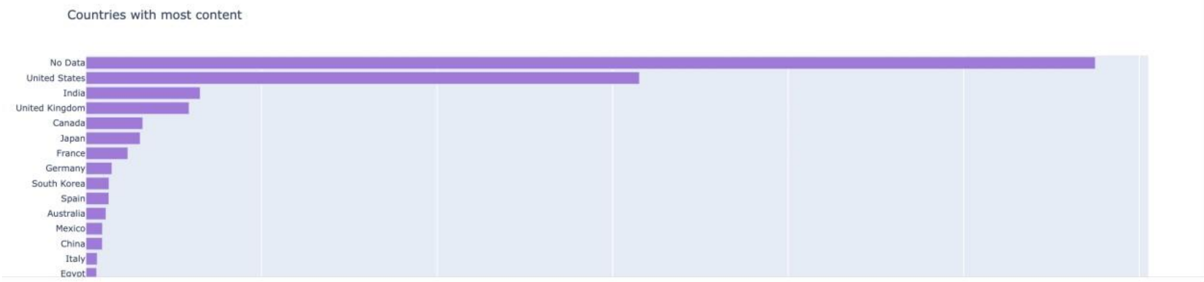
Top 10 Country by content type in country



4.5 Creating One unified Dataset and Visualizing it

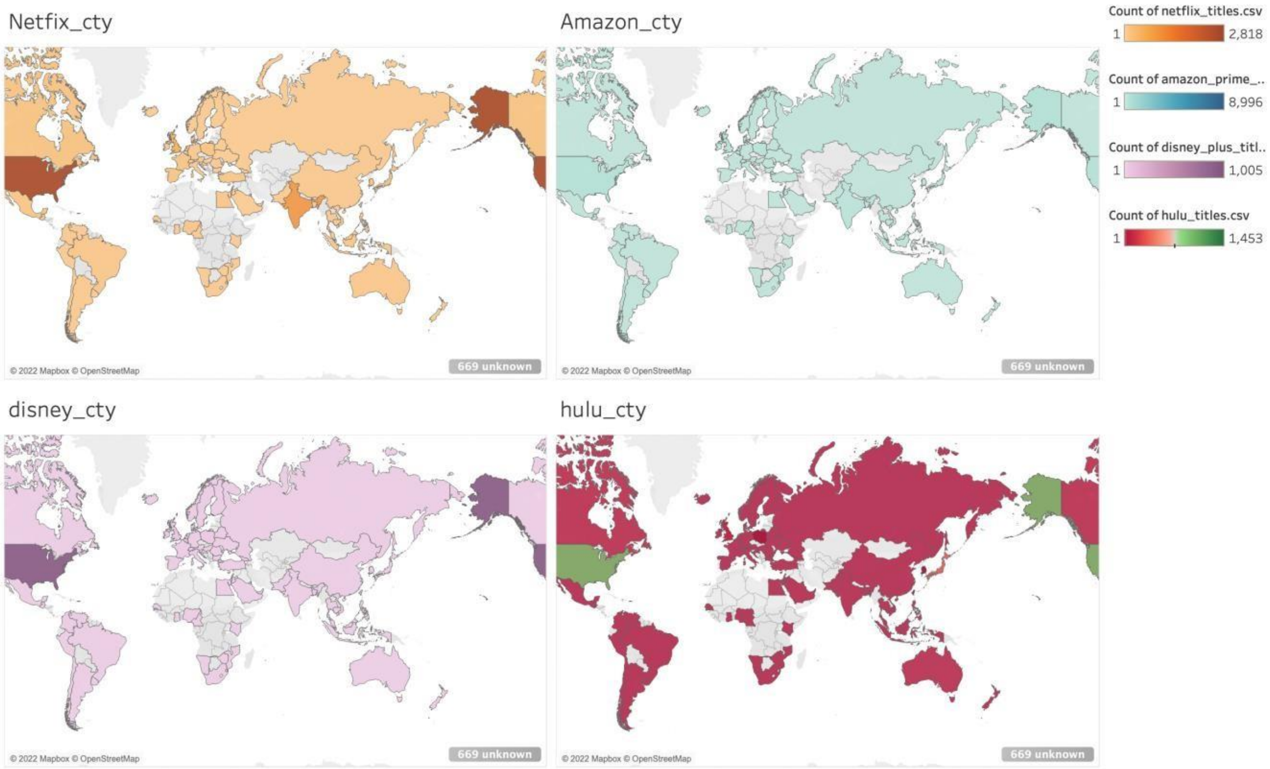

Movies by release year in major Streaming sites
Amazon Prime
Hulu
Disney
Netflix
1000
800
600
400
200
0
1920
1940
1960
1980
2000
2020


Distribution of release by platforms
Amazon
42%
38%
Netflix
13%
Hulu
6%
Disney

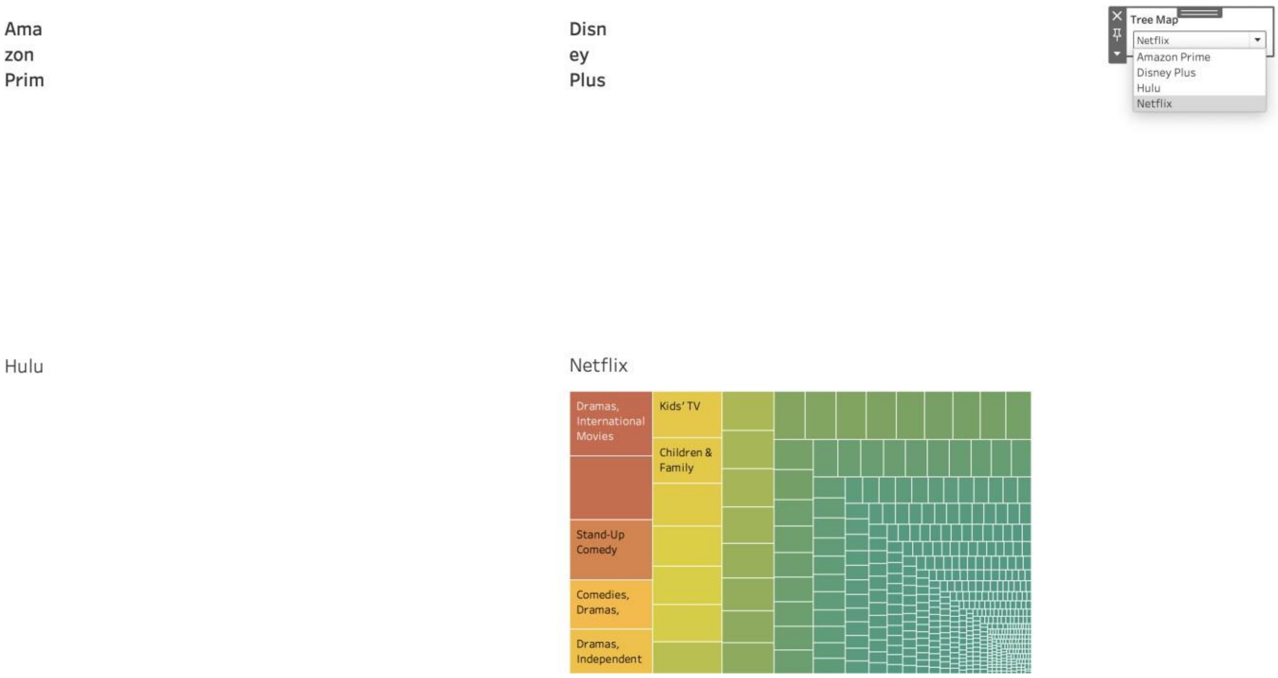


Interactive Plotly Graph in Python

4.6 Creating Tableau Dashboard



Dashboard displaying content by country in all OTT Platforms



Using Calculated Field to create dynamic Dashboard with a drop-down menu selector

4.7 Creating Clustering and Regression Model

```
[ ] def preprocessing(desc):
        desc = desc.lower()
        desc = re.sub('[-=+,#/\?:^$.@*\"*-&% · !』\\'|\(\)\[\]\<\>`\'…》]', ' ', desc)
        desc = " ".join(desc.split())

        return desc
```

```
result["new_description"] = result["description"].apply(lambda x: preprocessing(x))
print(result.shape)
result.head()
```

```
(22998, 13)
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	platform	new_description
Amazon	s1	Movie	The Grand Seduction	Don McKellar	Brendan Gleeson, Taylor Kitsch, Gordon Pinsent	Canada	2021	2014	No Data	113 min	Comedy, Drama	A small fishing village must procure a local d...	Amazon	a small fishing village must procure a local d...
	s2	Movie	Take Care Good Night	Girish Joshi	Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar	India	2021	2018	13+	110 min	Drama, International	A Metro Family decides to fight a Cyber Crimin...	Amazon	a metro family decides to fight a cyber crimin...
	s3	Movie	Secrets of Deception	Josh Webber	Tom Sizemore, Lorenzo Lamas, Robert LaSardo, R...	United States	2021	2017	No Data	74 min	Action, Drama, Suspense	After a man discovers his wife is cheating on ...	Amazon	after a man discovers his wife is cheating on ...
	s4	Movie	Pink: Staying True	Sonia Anderson	Interviews with: Pink, Adele, BeyoncÃ©, Britne...	United States	2021	2014	No Data	69 min	Documentary	Pink breaks the mold once again, bringing her ...	Amazon	pink breaks the mold once again bringing her c...
	s5	Movie	Monster Maker	Giles Foster	Harry Dean Stanton, Kieran O'Brien, George Cos...	United Kingdom	2021	1989	No Data	45 min	Drama, Fantasy	Teenage Matt Banting wants to work with a famo...	Amazon	teenage matt banting wants to work with a famo...

```
[ ] from gensim.models.fasttext import FastText as FT_gensim

    corpus = result["new_description"].tolist()
    sentences = [re.split(' ', str(sentence)) for sentence in corpus]
    print(corpus[0])
    print(sentences[0])
```

```
a small fishing village must procure a local doctor to secure a lucrative business contract when unlikely candidate and big city doctor paul lewis lands in their lap for a trial residence the townsfol
['a', 'small', 'fishing', 'village', 'must', 'procure', 'a', 'local', 'doctor', 'to', 'secure', 'a', 'lucrative', 'business', 'contract', 'when', 'unlikely', 'candidate', 'and', 'big', 'city', 'doctor
```

We first preprocess the description field to make it compatible with similarity checks

```
[ ] embedding_size = 30

    FT_model = FT_gensim(size=embedding_size, min_count=2, min_n=2, max_n=5, sg=1, negative=10,
                         sample=0.001, window=5, alpha=0.025, min_alpha=0.0001)

    FT_model.build_vocab(sentences)

    print('corpus_count: ', FT_model.corpus_count)
    print('corpus_total_words: ', FT_model.corpus_total_words)

    FT_model.train(sentences,
        epochs=FT_model.epochs,
        total_examples=FT_model.corpus_count, total_words=FT_model.corpus_total_words)

    print(FT_model)
```

```
corpus_count:  22998
corpus_total_words:  796584
FastText(vocab=22977, size=30, alpha=0.025)
```

```
[ ] FT_vector = []

    for item in corpus:
        FT_vector.append(FT_model.wv[str(item)])
    FT_vector = np.asarray(FT_vector)
```

```
[ ] from sklearn.cluster import KMeans
    from scipy.spatial.distance import cdist

    kmeanModel = KMeans(n_clusters=50, random_state=42).fit(FT_vector)
    cluster_id = kmeanModel.predict(FT_vector)
    result["cluster_id"] = cluster_id
```

Then we make clusters using the K-Means algorithm and appending the cluster id with the dataset.

The Data is divided into a total of 49 clusters.

```
[ ] result.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	platform	new_description	cluster_id
Amazon	s1	Movie	The Grand Seduction	Don McKellar	Brendan Gleeson, Taylor Kitsch, Gordon Pinsent	Canada	2021	2014	No Data	113 min	Comedy, Drama	A small fishing village must procure a local d...	Amazon	a small fishing village must procure a local d...	17
	s2	Movie	Take Care Good Night	Girish Joshi	Mahesh Manjrekar, Abhay Mahajan, Sachin Khedekar	India	2021	2018	13+	110 min	Drama, International	A Metro Family decides to fight a Cyber Crimin...	Amazon	a metro family decides to fight a cyber crimin...	49
	s3	Movie	Secrets of Deception	Josh Webber	Tom Sizemore, Lorenzo Lamas, Robert LaSardo, R...	United States	2021	2017	No Data	74 min	Action, Drama, Suspense	After a man discovers his wife is cheating on ...	Amazon	after a man discovers his wife is cheating on ...	7
	s4	Movie	Pink: Staying True	Sonia Anderson	Interviews with: Pink, Adele, BeyoncÃ©, Britne...	United States	2021	2014	No Data	69 min	Documentary	Pink breaks the mold once again, bringing her ...	Amazon	pink breaks the mold once again bringing her c...	37
	s5	Movie	Monster Maker	Giles Foster	Harry Dean Stanton, Kieran O'Brien, George Cos...	United Kingdom	2021	1989	No Data	45 min	Drama, Fantasy	Teenage Matt Banting wants to work with a famo...	Amazon	teenage matt banting wants to work with a famo...	0

Now, we create the recommendation system

```
[ ] def recommendation_system(title_name):
        top_k = 5
        title_row = result[result["title"] == title_name].copy()
        search_df = result[result["cluster_id"].isin(title_row["cluster_id"])].copy()
        search_df = search_df.drop(search_df[search_df["title"] == title_name].index)

        search_df["Similarity"] = search_df.apply(lambda x: FT_model.wv.similarity(title_row["new_description"], x["new_description"]), axis=1)
        search_df.sort_values(by=["Similarity"], ascending=False, inplace=True)

        return search_df[["title", "Similarity"]].head(top_k)
```

```
[ ] recommendation_system("Ernest Saves Christmas")
```

	show_id	title	Similarity
Netflix	s1557	A Trash Truck Christmas	[0.9858199]
Amazon	s9378	Noddy Saves Christmas	[0.98308843]
	s2658	Dino Dana The Movie	[0.9823687]
Netflix	s7319	Little Singham Bandarpur Mein Hu Ha Hu	[0.9823305]
Amazon	s1765	Magical Playtime with Mila and Morphle	[0.9812304]

```
[ ] recommendation_system("National Parks Adventure")
```

	show_id	title	Similarity
Netflix	s4052	2,215	[0.99168164]
Hulu	s490	Summer of Soul	[0.9916012]
Netflix	s5112	Myths & Monsters	[0.99127275]
	s682	They've Gotta Have Us	[0.9912634]
	s1917	Rize	[0.9911077]

Our Recommendation System takes a movie or show name as input and then narrows its search space to the cluster that they belong to. Then it runs a similarity check on the description of the entered title with every entry on the cluster.

It then returns a list of similar movies and which OTT platform you can watch that content.

CONCLUSION

From the Visualization we gained a lot of Inferences. Like how each platform values movies more than tv shows. We also found that Amzon and Netflix has the biggest content library with Disney & Hulu slowly building their catalogues. We also saw how US is the biggest producer of OTT Content with India coming at a close Second. We also inferred how the growth of OTT Content libraries has been meteoric in recent years, almost growing exponentially. We also saw the rating distribution between the OTTs and how they favor older teens/Adult markets as their main customer segment.

Finally, we created and tested the recommendation engine. We can see how such engines use clustering to reduce runtime dramatically while producing high quality results. This also highlights the importance of clustering data in large corporate environments like multinational OTT providers.

This proves that clustering isn’t just a mere visualization tool but also a very important machine learning implementation that reduces runtimes in such demanding worloads drastically.