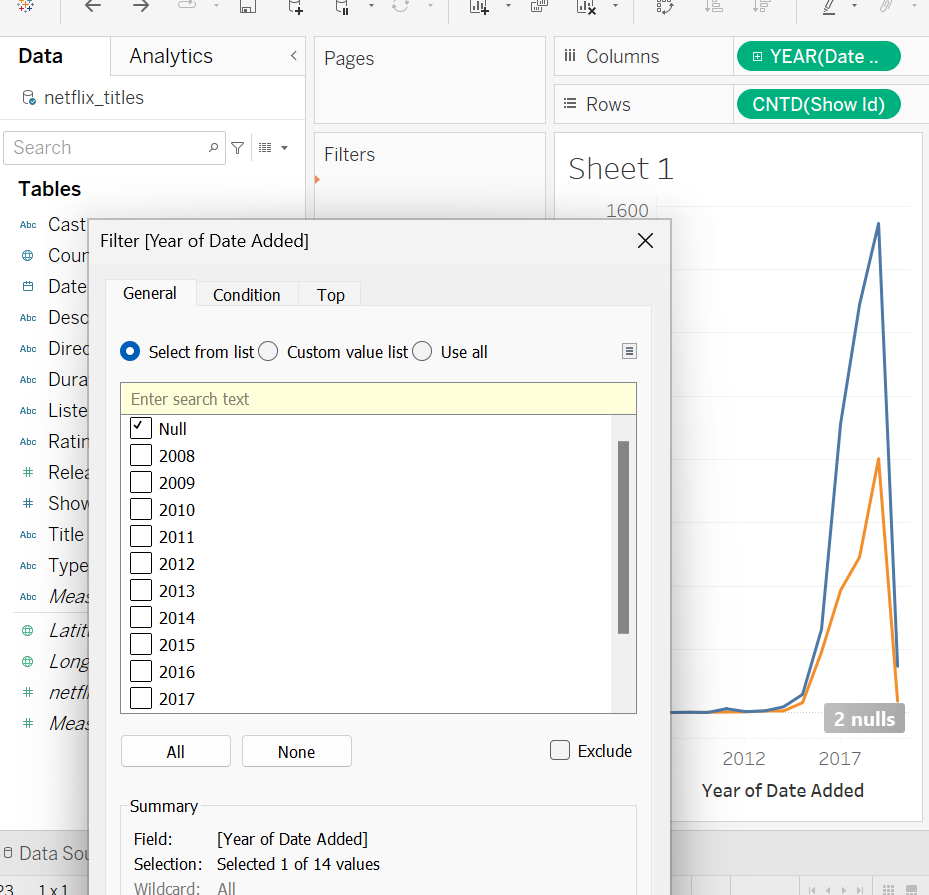
CST3340- Coursework 2

**Introduction**

The dataset that I will be using to analyse and visualise is a Netflix dataset that I acquired on GitHub. The dataset represents a collection of shows and movies available on Netflix, with each observation corresponding to a specific title. The variable available includes:

* **Cast:** this variable lists the actors and actresses who starred in the show or movie.
* **Country:** This variable indicates the country, or countries associated with the production or distribution of the show or movie.
* **Date Added:** This variable represents the date when the show or movie was added to the Netflix platform.
* **Description:** This variable provides a brief description/summary of the show or movie.
* **Director:** This variable lists the directors who worked on the show or movie.
* **Duration:** This variable specifies the duration/length of the show or movie.
* **Listed In:** This variable categorises the show or movie into one or more genres.
* **Rating:** This variable indicates the content rating assigned to the show or movie for example, PG, TV-MA, etc.
* **Released Year:** This variable represents the year when the show or movie was originally released.
* **Show ID:** This variable uniquely identifies each show or movie in the dataset.
* **Title:** This variable contains the title or name of the show or movie.
* **Type:** This variable indicates whether the entry is a TV show or a movie.

With this dataset, I had to clean it to so that it was ready to use.



***Figure 1 shows the removal of Null values.***

The image above shows how I filter out the dataset so that ‘Null’ will not be featured on the visualisation. Null values can introduce inaccuracies and inconsistencies in the dataset, affecting the overall quality of the data. So, by removing null values, it will ensure that the remaining data is reliable and suitable for analysis.

**Data Analysis and Visualisation**

A graph showing the amount of movies and tv shows

Description automatically generated

***Figure 2 shows the total Movies and TV shows by years.***

Figure 2 graph shows an area chart depicting the total Movies and TV by years. The colour shows details about ‘Type’, and the marks are labelled by ‘Type’. Since the dataset is very large, I filtered the ‘Date Added Year’ which excludes the ‘Null’ variable. On the graph, you can see the gradual increase of Movies and TV shows being added by 2015 and onwards, with 2019 being both its peak.

A screenshot of a computer screen

Description automatically generated

***Figure 3 shows the Total Movies and TV shows by Country.***

By using the map graph, the latitude and longitude was generated. The latitude and longitude filter keeps the non-null values. Where the sales are high, the concentration of the colour is higher, so where the colour of the country is more richer like the United States, is where the Movies and TV shows is most viewed around the world with the most being around 2,000.

A screenshot of a computer

Description automatically generated

***Figure 4 shows the Top 10 Genre.***

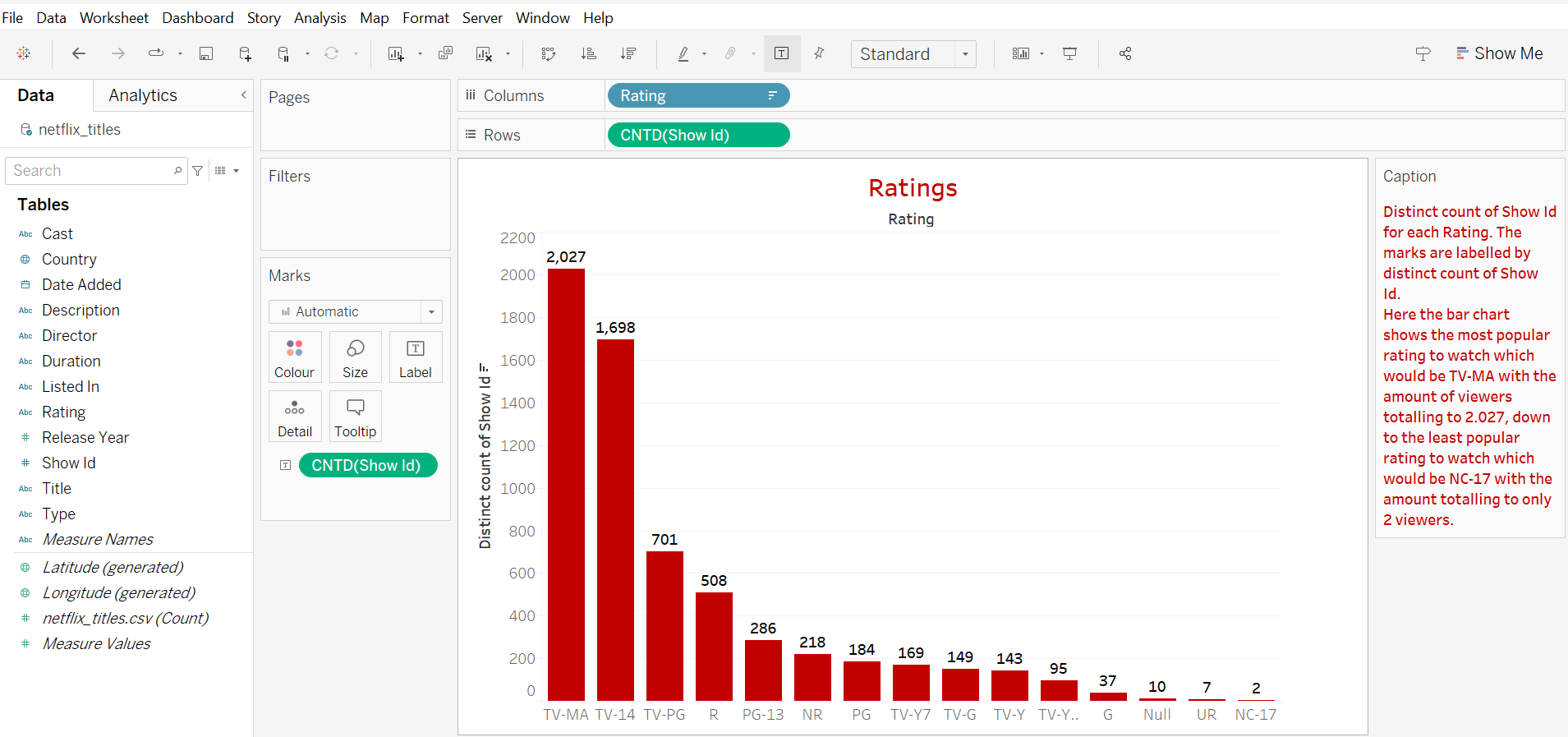
By using the horizontal bars, we can see the top 10 genre which was filtered on the ‘Listed In’ to keep only 10 out of the 461 genres. The horizontal graph above shows the data in ascending order from least popular to most popular genre where ‘Documentaries’ (299) are more favourited compared to ‘Children & Family Movies’ (120).

A screenshot of a computer

Description automatically generated

***Figure 5 shows Movie & TV Shows Distribution.***

By using the scatter plot graph we can see type, distinct count of show ID and the percentage of total distinct count of the show ID. The colour shows the details about the type, and the size shows the distinct count of the show ID. So, you can see the percentage of the ‘type’ Movie is 68.42% with the amount of show ID being 4,205 and next beside it is the ‘type’ TV show with the percentage 31.58% with the amount of show ID being 1,969. So, from the scatterplot we can see Movie having the most distinct show ID on the dataset.



***Figure 6 shows Ratings.***

Using the bar chart, it shows the most popular ratings to watch in descending order which would be TV-MA with the number of viewers totalling to 2.027, down to the least popular rating to watch which would be NC-17 with the amount totalling to only 2 viewers. Given that American television is the most diverse and that the majority of its content comes from networks like FX, HBO, Showtime, Adult Swim, and many more, it makes sense that TV-MA has the highest viewership. With NC-17 being the lowest viewed rating may be because its ‘Adults Only’ and no one 17 and under admitted and generally don’t get aired on TV.

A graph on a white background

Description automatically generated

***Figure 7 shows the trends of Genre.***

By using the line graph, we are able to see the trends of Genre through a period of time which differentiates itself with the top 10 genre (see figure 4). The colour shows details about ‘Listed In’ which keeps 10 of the 461 genres since the dataset is too big. The trend of genre shows the ‘listed in’ top 10 genre from the 1990’s to the late 2000’s with documentaries seen to have a huge increase in popularity in 2017 with the count of Netflix titles being 88 in contrast to Children & Family Movies, Comedies having only 10 counts of Netflix titles the same year.

A screenshot of a computer

Description automatically generated

***Figure 8 shows the Top 20 Directors and Movies.***

Using the stacked bars, I visualised the top directors and their movies contained on the dataset. The director filter shows the top 20 of 3,302 members, and the type was also filtered to only show movies. Figure 8 shows the top director in descending order being ‘Raul Campos and Jan Suter’ with the most directed films on the list with a count of 18 films, and ‘Don Michael Paul’ with the joint 20th spot with a count of 6 films on the list.

A screenshot of a computer

Description automatically generated

***Figure 9 shows Top 20 Cast and TV show.***

By using Horizontal stacked bars, we can see the top 20 cast and their featured TV show. The view is filtered on Type and Cast, with the type only keeping TV show, and cast being filtered to keep the top 20 of 5,470 members. The stacked bars include Cast, Release Year, Title, Type and Count of Netflix titles featured on the dataset. Figure 9 shows David Attenborough as top 1 cast out of 20 with a count of 14 shows.

A screenshot of a computer

Description automatically generated

***Figure 10 shows Top 20 Durated TV show.***

Using the tree map, we can see the top durated TV show. It showcases the Type, Duration, Title, and Count of the Netflix titles featured on the dataset. The duration filter keeps 20 of 201 members, the Title filter keeps 20 of 6,172 members, and the Type filter keeps only the TV show.

A screenshot of a computer

Description automatically generated

***Figure 11 shows the Top 20 Durated Movie.***

Using side-by-side bars, we can see the top durated movie. In ascending order Figure 11 shows the duration on the x-axis with their associated Movie titles above. The data is filtered on Type, which keeps Movie. The Title keeps 20 of 6,172 members and the view is filtered on Title and Duration, where Duration filter keeps 20 of 201 members.

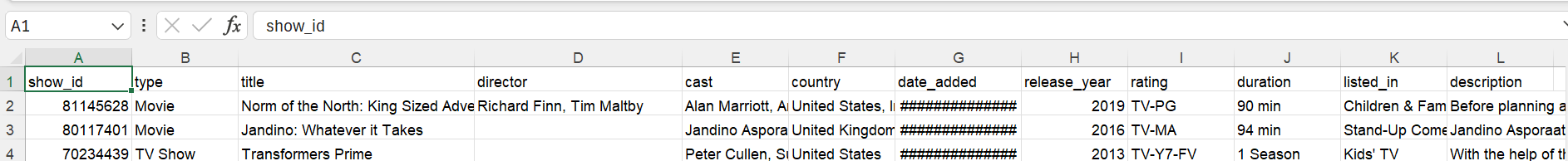
To conclude this section, the main findings I’ve found from the analysis of Netflix titles, is that TV-MA has a very high popularity when it comes to viewers watching tv shows and movies, and that the United States hold the most viewership in regards to the streaming service Netflix. This makes sense since Netflix is very popular in the States.

**Selection of Data Mining Algorithm and Data Pre-processing**

The data mining algorithm that is suitable for my data is clustering. Clustering algorithms, such as k-means, hierarchical clustering, and DBSCAN, are relatively simple to understand and implement but can be applied to a wide range of data mining tasks. The most common clustering algorithm is K-means. It works by selecting k initial centroids, where k is a user-defined number of clusters. Each data points are assigned to the closest centroid, and each collection of assigned points forms a cluster. The centroid of each cluster is then updated based on the members of the cluster. This process is repeated until the centroids no longer change significantly, indicating that the clusters are as consistent as possible (Sharma, 2024). This makes them a versatile tool in the data mining process, applicable to various domains like market research, pattern recognition, and image analysis. Weka also provides a comprehensive framework that integrates various clustering algorithms along with tools for data preprocessing, evaluation, and visualisation. This allows users like myself to easily experiment with different clustering techniques and evaluate their effectiveness on their specific data mining tasks.

The dataset contains features that could naturally form groups. For instance, movies and TV shows on Netflix can be clustered based genres, released years, or ratings, which could reveal trends in content distribution or popularity. With clustering, it can help identify these patterns without prior knowledge of the data’s grouping.

With opening up Weka and loading the dataset, I was met with errors with my csv file that was making it difficult to open up the dataset. This was resolved by removing the unique variable that was not useful for the Weka analysis.



***Figure 12 shows the original dataset before resolving the anomalies.***

***A screenshot of a computer

Description automatically generated***

***Figure 13 shows the csv file after resolving the anomalies.***

Looking at figure 12 , the unique variable that were removed was: show id, date added, listed in and description. Figure 13 shows the main variables for clustering which include:

* **Type:** This categorical variable (TV show/Movie) could help in distinguishing between two broad categories of content.
* **Genre**: categorical variable that can be transformed into several binary variables for clustering.
* **Release** **year**: Numerical variable that might help cluster content by era.
* **Country**: Categorical variable that could be used to cluster titles by their country of origin.
* **Rating**: Another categorical variable that might group titles by their audience suitability.
* **Duration**: For movies, this numerical variable could help distinguish between short films and feature-length movies.

After removing the anomalies on the csv file, some of the anomalies that could’ve been dealt with also could have been filling or removing missing values, especially in **‘director’** and **‘country’,** and transforming the **‘duration’**  into a uniform numerical format that can be compared across movies and TV shows.

Following on the previous point **‘duration’,** a transformation from the mixed format (e.g. 90 min, Season 1) to a numerical scale would be crucial. For TV shows, converting seasons to a numerical scale (e.g., assuming an average number of episodes or duration per seasons), making it comparable with the movie durations.

**Data Mining**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

***Figure 14 shows the implementation of the k-means clustering algorithm.***

Figure 14 shows k-means clustering algorithm by using the **“SimpleKMeans”** algorithm. The K-means algorithm is a method to partition n observation into k clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space into Voronoi cells.

A screenshot of a computer

Description automatically generated

***Figure 15 shows the configuration window.***

In the configuration window, various parameters can be set such as:

* **numClusters:** This is the number of clusters that is generated and is the **‘k’** in k-means.
* **distanceFunction:** The distance function used to measure the distance between instances. This is usually set to the default **Euclidean distance**.
* **maxIterations:** This is the maximum number of iterations to perform.

**First Iteration**

***A screenshot of a computer

Description automatically generated***

***A white paper with black text

Description automatically generated***

***Figure 20 shows the clusters obtained from the training and test set.***

***A computer screen shot of a computer script

Description automatically generated***

***Figure 21 shows the percentage split for testing split.***

Figure 20 shows the clustering of instances using training set, and based on the second iteration, I removed more variables from the dataset by changing the **‘numClusters’** to 2. With the number of iterations compared to the second iteration, it only 4 repeated times for it improve the accuracy and quality of the clustering. Which resulted to the time taken to build the model **‘0 seconds**’, as it’s shorter than the time taken for the full dataset, illustrating faster processing due to the reduced data volume.

**Sum of Squared Errors (SSE):** The SSE results shows **‘11466.272512475842’,** this value is relatively high and represents the total squared distance between each data point and its cluster centroid. A high SSE suggests that the clusters have greater variance, meaning that the data points are spread out more widely around the centroids. This is not very beneficial as it suggests that the clustering algorithm hasn’t effectively captured the natural groupings in the data. Other poor factors of having a higher SSE include inappropriate number of clusters, and presence of outliers etc.

**Comparing the percentage in each cluster between training and test set**

When running the second iteration, the **‘numClusters’** was changed to show the set amount of clusters which was 2. Figure 20 shows the clusters obtained from the training data which had 2408.0 in Cluster 0, and 1706.0 in Cluster 1. In the training set, there is 4114.0 cases.

The percentage of each cluster is:

* Cluster 0: 2408.0/4114.0\*100 = 58%
* Cluster 1: 1706.0/4114.0\*100 = 41%

Figure 21 shows the percentages given for the testing set and as the percentage in each cluster is similar for both the training and testing set, it also implies that the clusters are good.

**Second Iteration**A screenshot of a computer

Description automatically generated

A close-up of a computer code

Description automatically generated

***Figure 18 shows the clusters obtained from the training and test set.***

***A screenshot of a computer program

Description automatically generated***

***Figure 19 shows the percentage split for testing split.***

Figure 18 shows the clusters obtained from the training and test set and Figure 19 shows the percentage split for testing split. The number of iterations is 8 which means the algorithm repeated itself 8 times to try to improve the accuracy and quality of the clustering. With running the second iteration, more iterations were taken which can lead to better clustering, the time taken to build the model was **‘0.0 seconds.’**

**Sum of Squared Errors (SSE):** The SSE results shows **‘11086.086562727058’**, this specific value is the total sum of squared distances within all clusters of the dataset, and it’s a measure of the tightness of the clusters in the dataset. With this SSE, we understand that the more clusters we generate, the better and lower the values become with the clusters getting closer to their centroids.

**Comparing the percentage in each cluster between training and test set**

When running the second iteration, the **‘numClusters’** was changed to show the set amount of clusters which was 4. Figure 18 shows the clusters obtained from the training data which had 350.0 in Cluster 0, 654.0 in Cluster 1, 2024.0 in Cluster 2, and 1086.0 in Cluster 3. In the training set, there is 4114.0 cases.

The percentage of each cluster is:

* Cluster 0: 350.0/4114.0\*100 = 8%
* Cluster 1: 654.0/4114.0\*100 = 15%
* Cluster 2: 2024.0/4114.0\*100 = 49%
* Cluster 3: 1086.0/4114.0\*100 = 27%

Figure 19 shows the percentages given for the testing set and as the percentage in each cluster is similar for both the training and testing set, it also implies that the clusters are good.

**Third Iteration**

A screenshot of a computer

Description automatically generated

A screen shot of a television show

Description automatically generatedA screenshot of a computer

Description automatically generated

***Figure 16 shows the clusters obtained from the training and test set.***

A screenshot of a computer

Description automatically generated

***Figure 17 shows percentage split for the testing set.***

Figure 16 shows the clusters obtained from the training and test set and Figure 17 shows the percentage split for the testing split. When running the whole dataset, the number of iterations came to 7, this means the number of times the clustering algorithm repeatedly process the data to refine its results, which took **‘0.03 seconds’** to build the model. When using the percentage split, it means that the amount of data the algorithm uses to build the model is effectively reduced. For instance, the percentage split was set to 66% and will only use that percentage for the training and the remaining 34% for testing. The reduced data load naturally lead to quicker training times where it only took **‘0.03 seconds.’**

**Sum of Squared Errors (SSE):** The SSE results shows **‘10033.37487169592’** which tells us that this value is lower and indicates a better fit and that the instances within each cluster are closer to their centroids. Good factors of having a low SSE includes appropriate number of clusters, compact clusters and reduced impact of outliers.

**Comparing the percentage in each cluster between training and test set**

When running the last iteration, the **‘numClusters’** was changed to show the set amount of clusters which was 6. Figure 16 shows the clusters obtained from the training data which had 342.0 in Cluster 0, 387.0 in Cluster 1, 1090.0 in Cluster 2, 739.0 in Cluster 3, 1277.0 in Cluster 4, and 279.0 in Cluster 5. In the training set, there is 4114.0 cases.

The percentage of each cluster is:

* Cluster 0: 342.0/4114.0\*100 = 8%
* Cluster 1: 387.0/4114.0\*100 = 9%
* Cluster 2: 1090.0/4114.0\*100 = 26%
* Cluster 3: 739.0/4114.0\*100 = 17%
* Cluster 4: 1277.0/4114.0\*100 = 31%
* Cluster 5: 279.0/4114.0\*100 = 6%

Figure 17 shows the percentages given for the testing set and as the percentage in each cluster is similar for both the training and testing set implies that the clusters are good.

**Data Ethics**

Some of the **Ethical** considerations to consider relating to data analysis is the consent concerns. The dataset I’m using I obtained on GitHub where the creator of the dataset made it public on GitHub for everyone to use (DataScienceRoadMapDSRM, 2023). In regards to the privacy concerns no personal data was collected in the dataset.

The **Legal** considerations in regards to the GDPR is making sure it adheres to the data protection law, making sure the data process is fair and transparent. Also, purpose limitation, personal data must only be collected for a specific, explicit, and legitimate purpose.

In regards to the **Professional** Considerations it involves adhering to industry standards and codes of conduct that might not be legally required but considered best practice:

* **Accountability**: Taking responsibility for the results and choices made in light of the analysis, as well as for the ethical and legal ramifications of data analysis.
* **Integrity**: Preserving integrity by offering trustworthy, accurate data analysis and refraining from manipulating data to get the results you want.
* **Continuous education**: Staying current on changes to laws, technologies, and practices pertaining to data security and privacy.

**Conclusion**

**Visualisation**

In regards to the overall trends and patterns, Movies was seen to have more content in comparison to Tv shows, so the results of the rest of the visualisation were in favour to movies. An example of this can be seen on figure 4 where it showcases the top genres. The top genre is shown to be documentaries follow-up with stand-up comedy.

**Data mining**

In conclusion, the three iterations that were conducted yielded distinct patterns and trends in the primary findings. Running the dataset with a **‘numClusters’** of 2, and then increasing it to 4, and then finally iterating it to 6 resulted in a faster model construction time and tighter sum of squared errors clustering. My findings led me to believe that the more **‘numClusters’** that were increase led the SSE to improve by getting lower meaning that it’s a good clustering fit. To calculate the accuracy of the K-means, I used the percentage in each cluster between training and test set so that it can be compared. With my results the findings of all outcomes were accurate which allowed me to acknowledge the clusters being good.

In terms of business intelligence, these outcomes are essential. These insights can be used by managers and decision-makers to improve their strategic approaches. In particular, the clustering results can be used to pinpoint important market or customer segments that have similar characteristics. With the help of this data, products, marketing plans, and services can be better tailored to each segment's unique needs.

The results can also help allocate resources by showing which market segments are most profitable or have the greatest room for expansion. Subsequently, these high-potential groups' investments and initiatives could be given priority by decision-makers.

In summary, Weka's clustering techniques have not only produced a strong model that fits the data well, but they have also created a wealth of opportunities for improving business intelligence. By utilising these insights, businesses can make data-driven, better-informed decisions that are customised to the particulars of their clientele or working environment. In the end, their competitive advantage in the market will come from this strategic use of data.

# References

DataScienceRoadMapDSRM. (2023, February 27). *Tableau-Dashboards-info*. Retrieved from Github: https://github.com/DataScienceRoadMapDSRM/Tableau-Dashboards-info/blob/main/netflix\_titles.csv

Sharma, N. (2024, April 15). *K-Means Clustering Explained*. Retrieved from neptune.ai: https://neptune.ai/blog/k-means-clustering#:~:text=K%2Dmeans%20is%20a%20centroid,of%20groups%20in%20the%20dataset.