**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | Data Visualization Techniques, Machine Learning for Business |
| **Assessment Title:** | CA2 |
| **Lecturer Name:** | David McQuaid, Muhammad Iqbal |
| **Student Full Name:** | Daniela Mariano Barreto |
| **Student Number:** | 2023278 |
| **Assessment Due Date:** | 26/05/2024 |
| **Date of Submission:** | 26/05/2024 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

Table of Contents

[**Introduction** 3](#_Toc167570487)

[Recommendation System 3](#_Toc167570488)

[Word count 3](#_Toc167570489)

[**Business Understanding** 4](#_Toc167570490)

[**Data Understanding** 4](#_Toc167570491)

[Data description 4](#_Toc167570492)

[**Data Preparation** 5](#_Toc167570493)

[**Machine learning models** 7](#_Toc167570494)

[Content filtering 7](#_Toc167570495)

[Collaborative filtering 8](#_Toc167570496)

[User-User 8](#_Toc167570497)

[Item-Item 9](#_Toc167570498)

[Comparison 9](#_Toc167570499)

[Market Basket Analysis 9](#_Toc167570500)

[Apriori and Frequent Pattern (FP-Growth) 10](#_Toc167570501)

[Speed test: Apriori and FP-Growth 10](#_Toc167570502)

[**Interactive dashboard** 11](#_Toc167570503)

[Audience 12](#_Toc167570504)

[Link to access the interactive dashboard 13](#_Toc167570505)

[Rationale and justification for data preparation for visualizations 13](#_Toc167570506)

[**References** 13](#_Toc167570507)

# **Introduction**

With the advancement of the internet in people's lives, a large amount of data is generated, such as products researched, purchases made, films watched, music listened to, and more. Without a system that makes accurate recommendations most of the time, customers could get lost in the vast number of options and might even give up on making a purchase or watching a movie, which tends to impact the company's revenue. That is why many companies, such as Amazon, Netflix, and eBay, invest in recommendation systems.

## Recommendation System

Recommendation systems are algorithms designed to suggest relevant items to users based on the similarity of items or the characteristics of the user's profile (Patel, Patel and Chauhan, 2023, p.851). In other words, these algorithms can analyse user behaviour, interests, and characteristics to suggest similar products or services based on previous interactions.

This system interacts with users to learn their characteristics and preferences, storing this feedback in the recommender database that can be used for generating new recommendations for users with similar characteristics (Ricci et al., 2015, p.3; Patel, Patel and Chauhan, 2023, p.851).

According to Ricci et al. (2015, p. 5), there are several reasons to use a recommendation system in online retail businesses, such as increasing the number of items sold by tailoring to the user’s needs and wants, as well as selling more diverse items by offering items that might be hard to find, and so on. There are several techniques used to develop recommendation systems. In this project, the focus will be on Content-based and Collaborative-based filtering.

## Word count

|  |  |
| --- | --- |
| Introduction: | 236 |
| Business Understanding | 111 |
| Data Understanding | 153 |
| Data Preparation | 275 |
| Content / Collaborative | 630 |
| Apriori / FP-growth | 313 |
| : |  |
|  |  |
| Total: |  |

# **Business Understanding**

In this project, I used the Cross Industry Standard Process for Data Mining (CRISP-DM), a helpful method for managing data mining projects and making decisions.

To develop this project, I created a business in which I had to answer some questions to help stakeholders make decisions, such as:

*Background*

A company called AnimeNow has been in the streaming sector for a decade. Recently, with the increase in the popularity of anime, the company wants to evaluate its users' satisfaction to improve the recommendation systems.

*Business objectives*

To optimize user satisfaction and retention by enhancing the effectiveness of anime recommendations, thereby capitalizing on the growing popularity of anime within our streaming platform.

# **Data Understanding**

## Data description

*Recommendation systems*

The datasets used in this project contain records about anime, a distinctive style of animated shows or movies originating from Japan that has been gaining global appreciation and recognition (Binjola, 2023, p.1). The datasets are from the Kaggle repository (www.kaggle.com, n.d.) that was first gathered from myanimelist.net API (Link below). There are two datasets: one named ‘anime.csv’, which contains 12,294 records of various anime and 7 features; the second dataset is called ‘rating.csv’ and has 7,813,737 records and 3 features. The 'rating' dataset will be referred to as 'user' to facilitate understanding that these records are from the users. The data dictionaries are presented in Tables 1 and 2.

Dataset: [Anime Recommendations Database (kaggle.com)](https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database)

**Table 1:** Data dictionary of ‘anime.csv’ dataset.

A screenshot of a computer

Description automatically generated

**Table 2:** Data dictionary of ‘rating.csv’ dataset.

A screenshot of a computer

Description automatically generated

*Market basket analysis*

In this analysis, the dataset used is from Kaggle and contains e-commerce data from an online electronics store (Kabir, n.d. p.1). It is composed by 92,250 records and 5 features, such ‘product’ and ‘transaction\_id’. The data dictionary is presented in Table 3.

**Table 3:** Data dictionary of ‘ecommerce.csv’ dataset.

A screenshot of a computer

Description automatically generated

# **Data Preparation**

**Data cleaning**

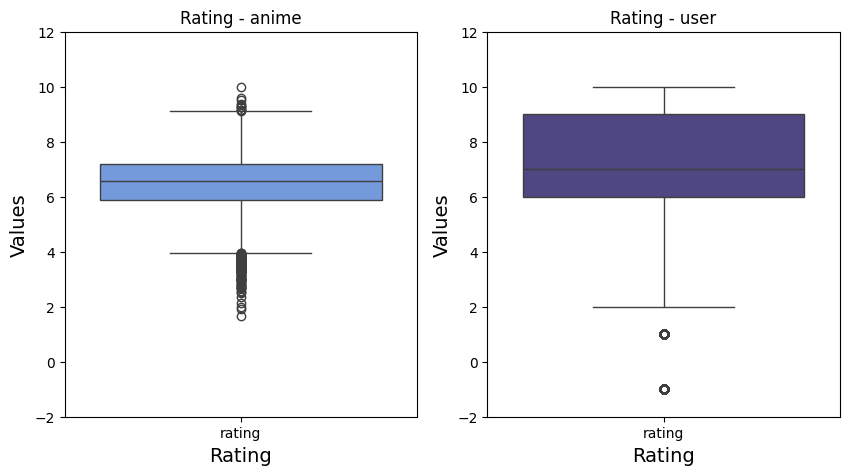
*Recommendation systems*

I started this step evaluating the data distribution through the descriptive statistics, as it is crucial for understanding the data's characteristics during the exploratory data analysis (EDA) phase. These insights will also aid in selecting the most appropriate method to handle missing values which is essential for ensuring data quality when performing statistical analysis, visualization, and modelling.

Evaluating the descriptive statistics in 'anime' dataset by comparing the difference between the mean and median (50%) values, I found that the data is slightly skewed in the feature 'rating', but highly skewed in the 'members' feature. This indicates that the data in these two columns do not follow a normal distribution. Similar happened with the 'user' dataset in which the 'rating' feature is slightly skewed, thus do not follow a normal distribution as well.

After search missing values, 2.25% of the 'anime' dataset was removed due to this proportion not being substantial. Regarding duplicates, only one was present and subsequently removed.

After searching for outliers, I found that they appear to be part of the dataset, as they fall within the expected range of rating values from 0 to 10. Because of this, I have decided to continue with them. The boxplot is presented in Figure 1.



**Figure 1:** Boxplot with outliers. a) ‘rating’ from ‘anime’ dataset, b) ‘rating’ from ‘user’ dataset.

b

a

*Market basket analysis*

The EDA and preprocessing were conducted on the 'ecommerce' dataset to perform Market Basket Analysis. With zero missing values and no duplicates, I proceeded to remove whitespace and cast the 'Transaction ID' to a string datatype to ensure consistency during the analysis. Additionally, I utilized one-hot encoding to transform the data into a format suitable for analysis with Apriori and Frequent Pattern Growth (FP-Growth) algorithms.

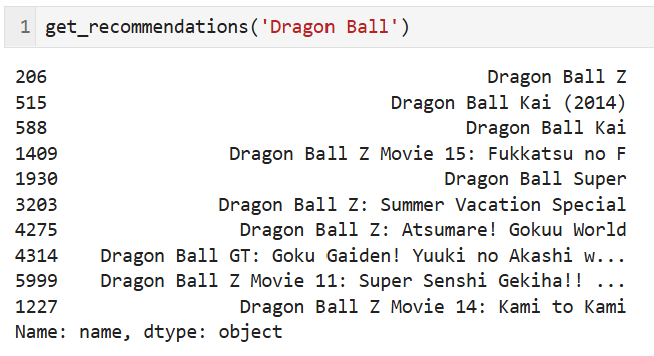
# **Machine learning models**

## Content filtering

This technique creates a user profile based on metadata (e.g., genre, actors, directors) provided either directly (explicit feedback) or indirectly (implicit feedback) by the user. Explicit feedback can include ratings given to a product, while implicit feedback can include browsing history or purchase behaviour. This information is then used to recommend products or services similar to those the user has shown a preference for (Patel, Patel, and Chauhan, 2023, p. 852).

In this project I applied the content-based filtering, using the anime genre to create a user profile and recommend anime. The result is presented in figure 2. We observe that if a user watches 'Dragon Ball', these anime titles would be recommended to the user. Based on the anime names, we can see that they are quite similar, suggesting good recommender performance.

**Figure 2:** Anime recommendations based on 'Dragon Ball'.



However, considering the dataset used, it cannot be considered good metadata, as there are a few features that limit the complexity and effectiveness of the recommendation system. Metadata is essential in a content-based recommender system, becoming even more important for making recommendations to new users and users with limited interactions. Despite its limitations, this data might still be useful for example usage.

## Collaborative filtering

Collaborative filtering is widely implemented, especially on e-commerce sites. This technique works by identifying similarities between user-item interactions and recommending items based on what similar users have liked or purchased in the past (Ricci et al., 2015, pp. 12-13). This system is more comprehensive because it relies on user behaviour rather than item metadata, which is the case in content-based filtering.

In collaborative filtering, a product can be recommended based on the preferences of similar users, which is called ‘user-user’ collaborative filtering. On the other hand, recommendations can be based on the historical preferences of the user, known as ‘item-item’ collaborative filtering.

Given the substantial size of the 'user' dataset, boasting over 7.8 million records, I have opted to narrow down to a specific genre, 'Thriller', for user-user and item-item collaborative filtering. This strategic move aims to streamline computational demands and processing time.

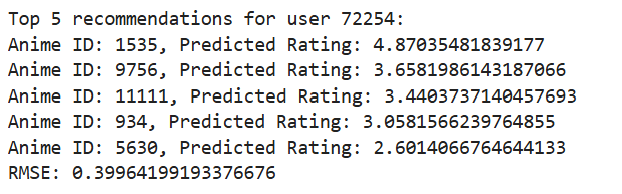
### User-User

For the user-user collaborative filtering I used the user-user matrix to compute the similarity between users through the Pearson correlation. This approach, normalizes the varying levels of generosity across their global rating patterns, thereby mitigating the influence of individual user biases (Aggarwal and Springer International Publishing Ag, 2018, p.34-38).

This recommender system considered only users who have a similarity score ('similarity\_threshold') of 0.8 or higher with user '72254', ensuring that the users have similar preferences to user '72254', potentially increasing the relevance of the recommendations (Figure 3). The predicted ratings were moderate to low for this user, possibly because the similar users do not have high ratings for these anime.

The root mean square error (RMSE) is relatively low (0.40), suggesting that the model's predictions are fairly close to the actual ratings. This value indicates that the predicted ratings deviate from the actual ratings by an average of 0.40 units.

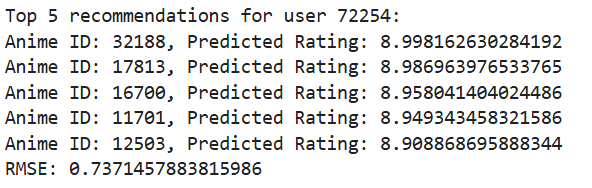
**Figure 3:** User-user recommendations for user ‘72254’.



### Item-Item

The cosine function was applied in item-item collaborative filtering to calculate the similarity between items and recommend anime to user ‘72254’ (Figure 4). The predicted ratings are around 9, suggesting that the user is predicted to highly enjoy these recommended anime. The RMSE resulted in an average of 0.74, indicating that the predictions are fairly accurate.

**Figure 4:** Item-item recommendations for user ‘72254’.



## Comparison

The comparison of recommendation systems showed that content-based filtering performed well with similar recommendations but is limited by inadequate metadata, making it less effective for new or low-interaction users. User-user collaborative filtering, using high similarity scores to user '72254', showed moderate to low predicted ratings with a low RMSE of 0.40, ensuring relevant recommendations. Item-item collaborative filtering predicted high enjoyment ratings around 9 for user '72254' with an RMSE of 0.74, indicating fairly accurate predictions. Overall, content-based filtering excels with rich metadata, user-user filtering benefits from user interaction data, and item-item filtering leverages item similarity for accurate and enjoyable recommendations, supporting a hybrid approach to improve recommendations.

## Market Basket Analysis

Also known as association-rule it is a method employed to unveil customer purchase patterns by analysing transactional data from stores. This insightful approach can yield a competitive edge for retail companies. By discerning the typical items a customer purchases, it facilitates strategic enhancements in store layouts, website design, and marketing strategies, such as promoting bundled offerings (Chen et al., 2005, p.339).

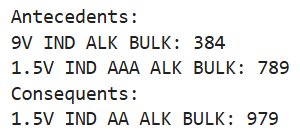
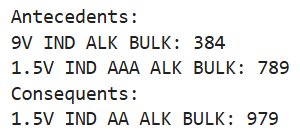
### Apriori and Frequent Pattern (FP-Growth)

Apriori and FP-Growth are the most common algorithms for mining frequent itemsets by defining the minimum support parameter for identify the frequent itemsets.

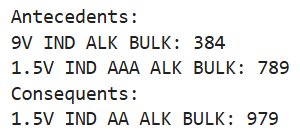
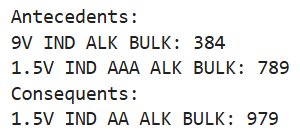
In this project, I used identical parameter values for both algorithms to ensure a reliable comparison between them. The results are presented in Figure 5. As we can see, the results are identical, thanks to maintaining the same parameters in both algorithms, despite each model having different approaches. The clear consistency found here suggests that they can identify the antecedents and consequents without ambiguity.

Therefore, we cannot express major similarity/divergence between these models when considering the output as both algorithms are deterministic, that means that they will result in the same output if the input and parameters are the same (Han, Pei and Yin, 2000). However, the difference between them lies in their efficiency and scalability, as we will see in the following section about the speed test between them.

**Figure 5:** Apriori and FP-Growth results.



Apriori



FP-Growth

### Speed test: Apriori and FP-Growth

After conducting a speed test, it is evident that the FP-Growth algorithm (1.36 seconds) is significantly faster than the Apriori algorithm (2.89 seconds), being 2.12 times quicker. This observation aligns with the findings of Hossain, Sattar, and Paul (2019). In addition, according to Heaton (2016), Apriori has serious scalability and memory issues compared to FP-Growth, making Apriori unsuitable for large datasets.

The superior performance of FP-Growth can be attributed to its use of the FP-tree structure, which minimizes database scans and reduces computational complexity. In contrast, Apriori relies on candidate generation and multiple database scans, which are computationally intensive (Shankar, 2024).



**Figure 6:** Speed test for Apriori and FP-Growth.

# **Interactive dashboard**

To create the interactive dashboard for this project, I chose to use nearly all the available features, given the limited number in the dataset. These features are essential for illustrating a business scenario for this category of streaming, which has been attracting the attention of major services like Netflix, now also offering this content. The features that I choose are:

* 'name'
* 'genre'
* 'type'
* 'episodes'
* 'rating'
* 'members'

The 'name' and 'rating' features indicate the most popular anime among the 'members'. Additionally, we can filter the data by 'gender'. The anime 'type' and the number of 'episodes' can be assessed based on the preferences of the 'members' as well. Overall, by analysing this data, we can gain insights into what users are watching and engaging with.

Based on the visualizations, I can conclude that this dataset is well-suited for machine learning models in the online retail business sector. The results align with identifying the most liked anime, taking into account the types and genres preferred by the members. This information can be utilized in machine learning to develop recommender systems, such as content-based and collaborative filtering, as implemented in this project.

This system is designed to suggest relevant items to users based on their behaviour or based on other users with similar profiles (Patel, Patel, and Chauhan, 2023, p. 851). This system can also provide initial recommendations for new users who do not have a history created, and it is usually done by recommending the top-rated items, such as the anime name and genre, which is shown in the interactive dashboard in this project.

## Audience

The interactive dashboard was designed for an audience of younger adults (18-35 years), requiring an understanding of their familiarity with technology, preference for clear, accessible information, and need for quick, direct answers (Desai, 2019). To provide direct information, I used the inverted pyramid structure, where the most important information is presented at the top, followed by less relevant information towards the bottom (Faisal, 2022).

With this in mind, I created an interactive dashboard using Streamlit, an open-source Python framework for creating interactive data apps with widgets like sliders and checkboxes, making it ideal for visualizing data and sharing insights (Streamlit, n.d.). I also used Plotly, a Python graphing library, to create interactive visualizations (Plotly, 2023). By using these two tools together, I believe the audience experience will be improved because they can easily hover over the chart to get values and select the information they want by using the menu box and the sidebar filters.

Considering that the audience is younger adults, I opted to use more vibrant colours to bring modernity and make the visuals more accessible for quickly seeing the information. I used a font size that can be read without effort, and I chose analogous colours (magenta, purple, and blue) to create balance in the dashboard and avoid overwhelming the audience.

At the end of the dashboard, I included a button to link the user directly to MyAnimeList, a website from which the data was gathered. This site also provides information related to the anime, such as synopsis, score, ranking, and more.

## Link to access the interactive dashboard

<https://dashboardanimepy-qy2t3nfjq3g2hc6pnylf5g.streamlit.app/>

## Rationale and justification for data preparation for visualizations

To create an interactive dashboard using Streamlit, I used the command ''%%writefile dashboard\_anime.py' in my Jupyter Notebook. This command is essential for writing the contents of a cell directly to a file (in this case, dashboard\_anime.py). This approach allowed me to make corrections and save the file without leaving the notebook. Additionally, I could see the final alterations in real-time by running the Streamlit environment from my Anaconda prompt.

After loading the preprocessed data, I used the code below to ensure that the 'episodes,' 'rating,' and 'members' variables were of numeric type to avoid errors during mathematical operations and filtering. The 'errors' parameter helped convert invalid parsing to NaN, ensuring that the data is clean.

A black text on a white background

Description automatically generated

Considering that the genres were separated by commas and there was more than one genre per row, I used the code below to separate them and count their occurrences. This allows filtering by individual genres.

****

I used the tools from Streamlit to create title, subheader with a white space and a divider:

**A close up of text

Description automatically generated**

I used the interactive sidebar filters from Streamlit for 'genre', 'type', 'episode', 'rating', and 'members' to facilitate the user experience when filtering for the desired values.

**A close-up of a text

Description automatically generated**

The use of filtering helps the user dynamically and efficiently select a subset of the data based on their criteria of interest. For example, the 'episodes', 'rating', and 'members' variables are filtered to check if their values fall within specified ranges. This ensures that only data matching the user's specified conditions is selected.

**A computer code with black text

Description automatically generated**

The code below is used to group the data by anime name and aggregate the values for 'episodes', 'rating', and 'members', allowing the analysis of individual anime titles ('name') rather than at the row level, which might contain multiple entries for the same anime name. The 'sum' and 'mean' function are used to summarize the data.

**A close up of text

Description automatically generated**

The code below defines a common layout for all the plots to ensure consistency among the charts.

**A white background with black text

Description automatically generated**

*Visualizations*

The first chart displays the top 10 anime based on their ratings. This type of chart provides a clear and straightforward comparison of the highest-rated anime.

**A close-up of a text

Description automatically generated**

The second visualization is a violin plot used to show the distribution of ratings across different genres, providing insights into how ratings vary within and across genres. A box plot is included inside the violin plot to adds additional summary statistics, such as the median and interquartile range.

**A computer code with black text

Description automatically generated with medium confidence**

The third visualization uses scatter plot to explore the relationship between anime ratings and the number of members (viewers). This visualization helps in identifying any correlation between popularity (as measured by members) and quality (as measured by rating).

A computer code with black text

Description automatically generated

The bubble chart is used to visualizes the relationship between ratings and the number of episodes, with the size of the bubbles representing the number of members. This three-dimensional aspect helps to understand how the number of episodes and members interact with ratings, revealing more complex patterns and trends.

A white background with black text

Description automatically generated

The strip plot (or dot plot) displays the distribution of ratings across different types of anime. This visualization helps in understanding how ratings are spread within each type and can highlight any outliers or patterns within specific types.

A close-up of text

Description automatically generated

The line chart with markers is used to show the average rating of anime based on the number of episodes. This visualization helps identify trends and patterns in how the length of an anime series (in terms of episodes) impacts its average rating, providing insights into viewer preferences regarding shorter versus longer series.

A computer code with text

Description automatically generated with medium confidence

The code below uses a Streamlit function to create a button that directs the user to the MyAnimeList website. I opted to do this because, considering the audience that is highly technological and appreciates quick information, I believe this is a way to provide it to them.



# **References**

Aggarwal, C.C. and Springer International Publishing Ag (2018). *Recommender Systems: The Textbook*. Cham Springer International Publishing Springer.

Patel, D., Patel, F. and Chauhan, U. (2023). Recommendation Systems: Types, Applications, and Challenges. *International Journal of Computing and Digital Systems*, 13(1), pp.851–868. doi:https://doi.org/10.12785/ijcds/130168.

Ricci, F., Lior Rokach, Bracha Shapira and Springerlink (Online Service (2015). *Recommender Systems Handbook*. New York, Ny: Springer Us.

www.kaggle.com. (n.d.). Anime Recommendations Database. [online] Available at: https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database [Accessed 13 May 2024].

Binjola, K. (2023). Effect of Anime on Personality and Popularity of Japanese Culture. In: Lecture Notes in Electrical Engineering. [online] Proceedings of International Conference on Recent Innovations in Computing. ICRIC 2022. Singapore: Springer. Available at: https://doi.org/10.1007/978-981-99-0601-7\_50 [Accessed 17 May 2024].

Kabir, F. (n.d.). Market Basket Analysis. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/farjanakabirsamanta/analytics-case-studyecommerce [Accessed 21 May 2024].

Chen, Y.-L., Tang, K., Shen, R.-J. and Hu, Y.-H. (2005). Market basket analysis in a multiple store environment. *Decision Support Systems*, 40(2), pp.339–354. doi:https://doi.org/10.1016/j.dss.2004.04.009.

Han, J., Pei, J. and Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, 29(2), pp.1–12. doi:https://doi.org/10.1145/335191.335372.

Hossain, M., Sattar, A.H.M.S. and Paul, M.K. (2019). Market Basket Analysis Using Apriori and FP Growth Algorithm. [online] *IEEE Xplore*. doi:https://doi.org/10.1109/ICCIT48885.2019.9038197.

Shankar (2024). Apriori vs. FP-Growth: Comparative Study of Data Mining Algorithms - Algorithmic Mind. [online] Algorithmic Mind. Available at: https://algorithmicmind.org/difference-between-apriori-and-fp-growth/ [Accessed 22 May 2024].

Heaton, J. (2016). Comparing dataset characteristics that favor the Apriori, Eclat or FP-Growth frequent itemset mining algorithms. [online] *IEEE Xplore*. doi:https://doi.org/10.1109/SECON.2016.7506659.

Faisal, F. (2022). Site is undergoing maintenance. [online] Aufait UX - Award winning UI UX Design Company. Available at: https://www.aufaitux.com/blog/dashboard-design-principles/ [Accessed 22 May 2024].

Desai, R. (2019). UX Design for Different User Generations. [online] Medium. Available at: https://uxplanet.org/ux-design-for-different-user-generations-a1eac5b8e403#:~:text=UX%20Design%20for%20Different%20User%20Generations%201%201. [Accessed 20 May 2024].

Streamlit (n.d.). Streamlit Docs. [online] docs.streamlit.io. Available at: https://docs.streamlit.io/ [Accessed 22 May 2024].

Plotly (2023). Plotly Python Graphing Library. [online] plotly.com. Available at: https://plotly.com/python/. [Accessed 22 May 2024].