UNIVERSITI MALAYA

WQD7001 Principles of Data Science

Group 11

Group Project Part 2

Name	**Matric Number**
Nasir Uddin Ahmed	S2015449
Muhammad Shahzad Rafiq	S2150889
Qixiang Yin	S2150692
Low Boon Kiat	17138399

Table of Contents

1. Team Members & Roles

Team members of Group 11 are listed in the table below:

Name	**Role**	**Link to E-Portfolio**
Nasir Uddin Ahmed	Leader, Detective	https://turjo7.github.io/PODSP/
Muhammad Shahzad Rafiq	Maker	https://drive.google.com/drive/folders/1N9omuJPiXIvuB1MaVVuNRxkmdKfbOxYT?usp=sharing
Qixiang Yin	Oracle	https://drive.google.com/drive/folders/1LRA72ofBMVxH0dk8bCLQilPMvQiY2llE
Low Boon Kiat	Presenter	https://drive.google.com/drive/folders/1pNAE1HyH5i0V6Sn8NTKYBdY2E7W7qwQL?usp=sharing

2. Executive Summary (from the Proposal)

Streaming media such as Netflix and Amazon Prime Video are gaining popularity because they provide flexibility for users to watch their favorite TV shows and movies at any time on any device. In November 2019, Disney released its own streaming service Disney+, claiming it as an active competitor in the streaming media industry. A vast range of on-demand content is available on Disney+ including films, TV series and documentaries for many popular brands such as Disney, Pixar, Marvel, Star Wars and National Geographic. Currently, over 160 million users worldwide subscribe to Disney+.

Humans tend to get overwhelmed quickly and make poor choices when presented with many options. This phenomenon, known as paradox of choice, applies to streaming media where users are presented with thousands of video content. According to consumer research, a typical streaming media user loses interest after spending 60 to 90 seconds (about 1 and a half minutes) reviewing 10 to 20 titles on one or two screens with chances of only 3 titles being reviewed in detail. At this point, the users will either find something of interest or the chances of them abandoning the streaming platform increases significantly (Gomez-Uribe & Hunt, 2015). This finding shows people hate choices because they are afraid of taking risks involved in decision making. This is where a recommendation system should be brought into play. It helps users to choose and makes the decision-making process easier for users.

Three objectives were formulated for this project. First is to build a recommendation system for Disney+ shows and movies that is accurate, efficient, and scalable. Second is to evaluate the performance of the recommendation system using a variety of metrics. Third is to deploy the recommendation system to production. The domain of this project is application of data science in streaming media industry. Today, the streaming media industry is rapidly growing. Personalizing user experience through recommendation systems has proven to help streaming media platforms to grow the number of subscribers. The scope of this project focuses on building a recommendation system that helps users to choose Disney+ video content.

The complete data science pathway has been applied to complete this data science project successfully. First, it is crucial to specify the project's goal and problem statement. This initial phase includes researching the domain and conducting background research. The following step is data collection and pre-processing which entails obtaining, cleaning, formatting and understanding the data to ensure the caliber and interoperability of the data. When the data is ready, exploratory data analysis is applied to obtain insights and identify patterns by using visualizations and statistical techniques. Next step is to apply machine learning algorithms and statistical models for modelling to develop a recommendation system. Once the model has been validated, an evaluation of its performance and generalizability is conducted. The last phase is to effectively communicate the outcomes and conclusions to stakeholders via visualizations, reports and presentations.

The dataset is obtained from Kaggle. It contains information for 992 shows available on Disney+. It includes 19 attributes such as IMDb ID, title, plot, type, rated, year, release date, runtime, genre, director, writer, actors, language, country, awards, Metascore, IMDB rating, and IMDB votes. Among the 19 attributes, 17 of them are categorical and 2 of them are numerical. For each attribute, missing values were identified. Mean value was used to impute the missing values for numerical variables. The most frequent value was used to impute the missing values for categorical variable. Next step is to identify duplicate rows and outliers in the dataset. 74 observations were found to be duplicate entries and subsequently dropped from the dataset. To handle outliers, a technique named winsorization was applied to replace extreme values with the nearest values within a specified range. To uncover hidden trends and useful insights, exploratory data analysis was conducted via descriptive statistics and visualization. Several charts were plotted including correlation heatmap, histogram, bar chart, word cloud, box plot, pair plot, pie chart and tree map. In this report, we will continue with modeling and deployment.

3. Mechanics- Hardware & Software & Platform Used

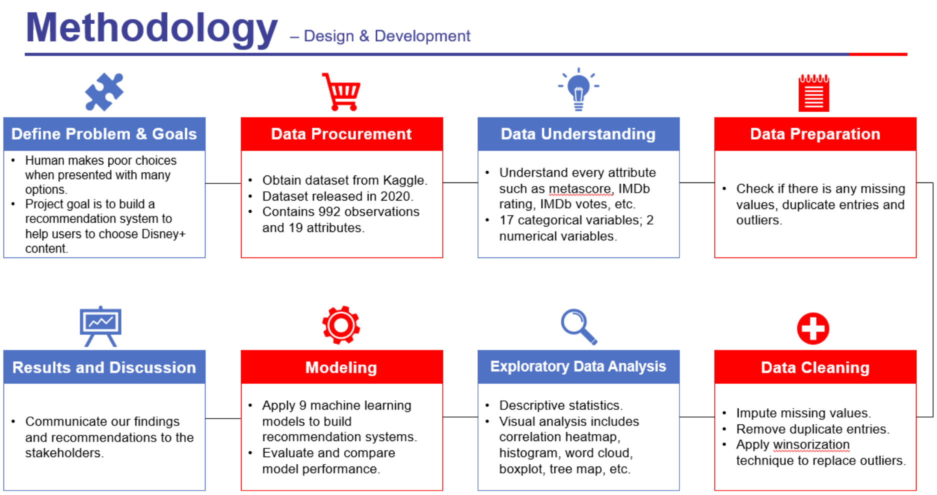
There were several hardware, software and platform used in completing this project. In terms of hardware, we used our own laptops to work on the project tasks namely doing research, formulating project scope, data manipulation, modeling, and reporting. According to Khan & Rajput (2023), the minimum hardware requirement for Python, our selected programming language, is as follows. The operating system suitable for Python includes Windows 7 and Mac OS X 10.11 or newer. CPU architecture that is compatible for Python includes Intel Core i3 or AMD Ryzen 3250u (64-bit). The minimum RAM and amount of hard disk required for running Python is 1GB RAM and 2GB hard-disk space.

In terms of software, we used Python as the programming language in completing this project. Python is chosen because we are more familiar with the tool as compared to other types of programming languages. Additionally, Python has vast number of open-source libraries such as pandas, NumPy, matplotlib and scikit-learn. These libraries make coding simpler and save our time and effort on data manipulation and modelling. Moreover, Python is one of the most accessible programming languages available because it has simple syntax which is easy to learn and understand. Due to its ease of learning and usage, python codes can be easily written using fewer lines, making it much simpler to use on many levels.

In terms of platform, we used Google Colab, Shiny and GitHub in completing this project. Google Colab is a useful cloud-based platform that allows us to collaborate with each other in writing Python codes. We created a Colab notebook and shared among ourselves for codes writing and execution. This Colab notebook is a suitable collaboration platform because it can be opened by multiple users at a time. If one person makes a change, the others will be able to see the change after a short delay. On the other hand, Shiny was used as a deployment platform to create interactive web applications. Shiny makes it easy to build interactive web apps straight from our code. It enables us to customize the layout and style of our application and dynamically respond to events, such as a button press, or dropdown selection. Shiny was chosen for our project because it is simple to use and does not require much web development knowledge. On top of that, we used GitHub as a code hosting platform for version control and documentation. By using Git repository, we can store, track and control changes to our code and data. This makes our documentation more organized and enables others to reproduce the results of our project from anywhere at any point of time. GitHub was chosen because reproducible research is an essential element of a good data science project in which reproducibility ensures the reliability of our findings.

4. Methodology - Design & Development

In methodology part, basic we will introduce about our project design and development part, this part we have 8 steps to do the design. Firstly, we specific our project problem and goal, our Project goal is to build a recommendation system to help users to choose Disney+ content due to when people facing too many options, they are hard to make decisions. Secondly, is data procurement, for this project we collect dataset from Kaggle website, this dataset is released in 2020 and containing 992 observations and 19 attributes. The third step is data understanding. We must understand every attribute such as metascore, IMDb rating, IMDb votes, etc. In this part, it totally included 17 categorical variables and 2 numerical variables. The fourth step is data preparation, the task is to check the missing values and duplicate entries and outliers. Then we go to the data cleaning, the main point of this part is to try imputing missing values, remove duplicate entries and we are trying to apply winsorization technique to replace outliers. The sixth step is going to EDA (Exploratory Data Analysis), mostly we are using descriptive statistics, also we applied visual analysis includes correlation heatmap, histogram, word cloud, boxplot, tree map, etc. The seventh step is through to the modeling, we are trying to use 9 kinds of machine learning models to build this recommendation system, beside that we evaluate and compare those models' performance. The last step is the result and discussion part, the main goal for this step is to communicate with our stakeholders with our findings and recommendations.



Methodology – Design & Development

Define Problem & Goals

- Human makes poor choices when presented with many options.
- Project goal is to build a recommendation system to help users to choose Disney+ content.

Data Procurement

- Obtain dataset from Kaggle.
- Dataset released in 2020.
- Contains 992 observations and 19 attributes.

Data Understanding

- Understand every attribute such as metascore, IMDb rating, IMDb votes, etc.
- 17 categorical variables; 2 numerical variables.

Data Preparation

- Check if there is any missing values, duplicate entries and outliers.

Data Cleaning

- Impute missing values.
- Remove duplicate entries.
- Apply winsorization technique to replace outliers.

Exploratory Data Analysis

- Descriptive statistics.
- Visual analysis includes correlation heatmap, histogram, word cloud, boxplot, tree map, etc.

Modeling

- Apply 9 machine learning models to build recommendation systems.
- Evaluate and compare model performance.

Results and Discussion

- Communicate our findings and recommendations to the stakeholders.

Figure 1: Methodology Followed in the Project

By following this methodology, the research project aims to develop and implement an innovative recommendation system for Disney+. This system will leverage machine learning algorithms and a carefully prepared dataset to assist users in making informed content choices, improving their overall streaming experience on the platform.

5. Experiment & Results

In this section, we present the experiments conducted and the results obtained during the development of the Disney+ Recommendation System. The aim of our group project is to explore and implement various recommendation techniques, including content-based filtering, machine learning algorithms, and collaborative filtering, to provide personalized recommendations to Disney+ users. And find the best model for the recommendation system.

We conducted a series of experiments to build and evaluate the performance of our recommendation system. We used a dataset of Disney+ movies and TV shows, with each item in the dataset having a set of features, such as genre, cast, imdb_rating, imdb votes, etc. Our experiments and results are summarized in the point form below.

5.1 Experiment

5.1.1 Data Collection and Preprocessing

We started by collecting a comprehensive dataset from Disney+ that included information about movies, TV shows, genres, cast, crew, ratings, and user interactions. The data was preprocessed to handle missing values, normalize numerical features, and convert categorical variables into a suitable format for analysis.

5.1.2 Content-Based Filtering

In the context of a streaming platform like Disney+, content-based filtering involves examining the textual information associated with movies or TV shows, such as titles, descriptions, genres, cast, crew, and other metadata. The system creates a profile or representation of each item by extracting relevant features from its content.

To implement content-based filtering, we leveraged the textual features of the movies and TV shows, such as titles, descriptions, genres, and cast. We performed the Bag-of-Words (BoW) approach with cosine similarity for content-based recommendation. BoW representation is a straightforward and intuitive approach. It treats documents as a collection of words and captures their frequency in a document. It does not require complex linguistic analysis or consideration of word order, making it easy to implement. BoW representation provides interpretable features. Each dimension in the vector represents a unique word, allowing you to understand which words contribute more to the similarity or dissimilarity between documents. This interpretability can be valuable in understanding the recommendations made by the system.

By incorporating features—genre, actors, plot, and title—the content-based filtering approach can capture distinct aspects of user preferences, such as preferred genres, favorite actors, storyline preferences, and linguistic or thematic similarities. Together, these features provide a rich representation of the content, enabling the recommendation system to generate personalized and relevant recommendations based on the users' previous interactions and preferences.

5.1.3 Traditional Machine Learning Approaches

To forecast recommendations based on the data, we used a variety of machine learning methods, including decision trees, random forests, lasso, ridge regression, bagging, etc. These models employed imdb_rating, imdb votes, cast, and crew information as inputs. To enhance model performance and reduce overfitting, we used cross-validation, train-test split, and hyperparameter adjustment. The result will be discussed in the upcoming section.

5.1.4 Collaborative Filtering

There are several types of Collaborative Filtering Approaches like User-based collaborative filtering, Item-based collaborative filtering, Hybrid collaborative filtering. They are described below.

User-based collaborative filtering: This algorithm recommends items to a user based on the ratings of other users who have similar interests.

Item-based collaborative filtering: This algorithm recommends items to a user based on the ratings of other items that the user has rated highly.

Hybrid collaborative filtering: This algorithm combines user-based and item-based collaborative filtering to improve the accuracy of recommendations.

Approach to **item-based collaborative filtering** can be effective even when there is no user data available. However, it is important to note that the accuracy of the recommendations will depend on the quality of the ratings data that is available. That is why we choose **item-based collaborative filtering.** We used Item- base collaborative filtering with **KNN (k-nearest neighbors) algorithm** to generate movie recommendations.

5.1.5 Evaluation Metrics

We used standard assessment metrics, including accuracy, recall, F1-score, mean average precision (MAP), R2, and MSE (Mean Squared Error), to assess the effectiveness of our recommendation system. To construct these measures, we randomly divided the dataset into training and testing sets, and we contrasted the system's recommendations with the real user preferences.

5.1.6 Data Product

For the data product, we developed an interactive web application using R Shiny. The application integrated the recommendation system and provided an intuitive user interface for users to explore personalized recommendations based on their preferences and interactions.

5.2 Results

5.2.1 Results- Content-Based Filtering

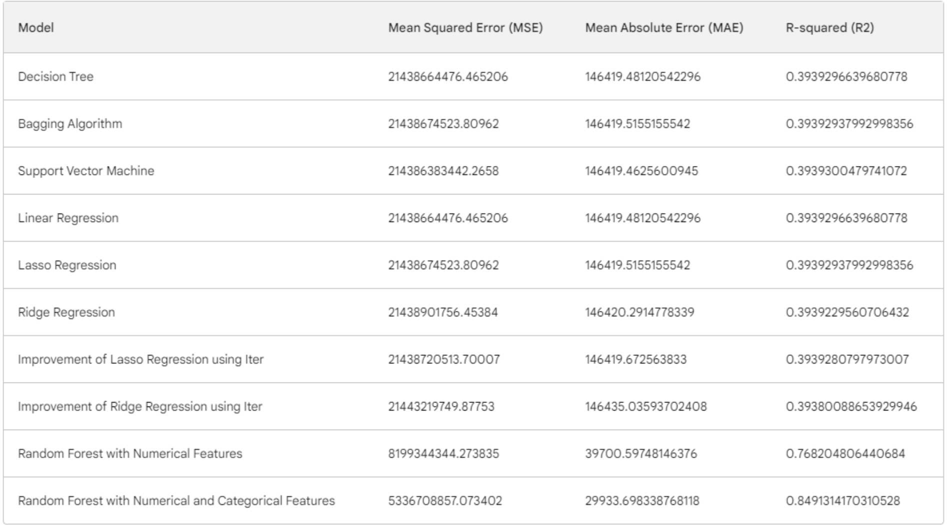
For content-based Recommender it is difficult to measure accuracy, and precision because here user rating is not available in our dataset.

The key difference between user rating and IMDb rating is the source of the ratings. User ratings come directly from individual viewers, capturing their individual opinions and preferences, while IMDb ratings are based on the aggregated ratings from registered IMDb users. User ratings tend to be more subjective and diverse, while IMDb ratings aim to provide a more standardized measure of a title's overall rating.

Content-based filtering is used to recommend movies based on their metadata such as genre, actors, and plot. The code constructs a "metadata soup" by combining the relevant metadata features for each movie. It then applies a Count Vectorizer to convert the text data into a numerical representation. Cosine similarity is computed on the count matrix to measure the similarity between movies based on their metadata.

5.2.2 Results- Machine Learning Approaches

We applied several machine learning models. The results of the approaches are shown below.



Model	Mean Squared Error (MSE)	Mean Absolute Error (MAE)	R-squared (R2)
Decision Tree	21438664476.465206	146419.48120542296	0.3939296639680778
Bagging Algorithm	21438674523.80962	146419.5155155542	0.39392937992998356
Support Vector Machine	214386383442.2658	146419.4625600945	0.3939300479741072
Linear Regression	21438664476.465206	146419.48120542296	0.3939296639680778
Lasso Regression	21438674523.80962	146419.5155155542	0.39392937992998356
Ridge Regression	21438901756.45384	146420.2914778339	0.3939229560706432
Improvement of Lasso Regression using Iter	21438720513.70007	146419.672563833	0.3939280797973007
Improvement of Ridge Regression using Iter	21443219749.87753	146435.03593702408	0.39380088653929946
Random Forest with Numerical Features	8199344344.273835	39700.59748146376	0.768204806440684
Random Forest with Numerical and Categorical Features	5336708857.073402	29933.698338768118	0.8491314170310528

Figure 2: Model Evaluation

For the classification models (**Decision Tree and Bagging Algorithm**), the accuracy is relatively low, indicating that the models are not performing well in terms of correctly predicting the classes. The precision, recall, and F1 score are also moderate.

Support Vector Machine (SVM) performs better than the classification models, with higher accuracy, precision, and recall.

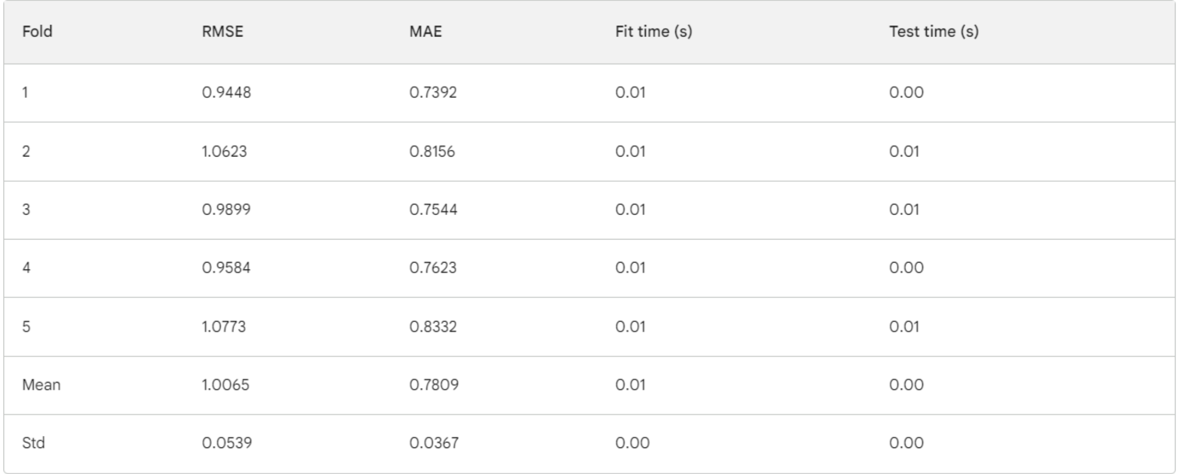
For the regression models (Linear Regression, Lasso Regression, Ridge Regression, Improvement of Lasso Regression using Iter, Improvement of Ridge Regression using Iter), the R-squared values are quite low, indicating that the models are not explaining a massive portion of the variance in the data. Additionally, the mean squared error (MSE) and root mean squared error (RMSE) are quite high, suggesting that the models have large prediction errors.

The **Random Forest models (with Numerical Features and Numerical and Categorical Features)** perform the best among the models listed. They have the lowest mean squared error (MSE), mean absolute error (MAE), and highest R-squared value. These models provide the best predictive performance, considering both numerical and categorical features.

Based on these observations, the **Random Forest model with Numerical and Categorical** Features appears to be the best model among the options provided for your movie recommender system. It demonstrates the lowest prediction errors and the highest R-squared value, indicating a better fit to the data and potentially better predictive performance.

5.2.3 Results- Item - based Collaborative Filtering

We used Item- base collaborative filtering with KNN (k-nearest neighbors) algorithm to generate movie recommendations. The algorithm is evaluated using cross-validation with 5 folds. The evaluation metrics used are Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).



Fold	RMSE	MAE	Fit time (s)	Test time (s)
1	0.9448	0.7392	0.01	0.00
2	1.0623	0.8156	0.01	0.01
3	0.9899	0.7544	0.01	0.01
4	0.9584	0.7623	0.01	0.00
5	1.0773	0.8332	0.01	0.01
Mean	1.0065	0.7809	0.01	0.00
Std	0.0539	0.0367	0.00	0.00

Figure 3: Item – based Collaborative Filtering Model Evaluation

As we can see, the mean RMSE and MAE are both good, but there is still some variation between the folds. This suggests that the recommendation system is not yet fully converged. However, the overall results are promising, and there is potential for further improvement.

So, in future by using a larger dataset, more sophisticated algorithm, hybrid approach, personalization, we can improve it.

5.2.4 Results - Summary

After analyzing multiple models for a movie recommendation, the Random Forest algorithm was selected as the most suitable choice. Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. In the context of movie recommendation, Random Forest utilizes various features such as runtime, IMDb rating, Metascore, genre, director, country, and actors to predict IMDb votes. It is particularly effective in handling both numerical and categorical data, which makes it suitable for movie-related attributes. Random Forest considers the interactions between distinctive features and produces accurate predictions by aggregating the results from multiple decision trees. Its ability to handle complex relationships and provide feature-importance insights makes it a reliable and robust algorithm for movie recommendation systems.

6. Plan for Reproducible Research

Reproducible research is a scientific practice that aims to make research findings transparent, accessible, and verifiable. By following the principles of reproducible research, researchers can increase trust in their findings and make it easier for others to build upon their work.

We believe that reproducible research is an important part of scientific practice. By following the principles of reproducible research, we can increase the transparency, accessibility, and verifiability of our research findings. Here is a more detailed breakdown of each step of our future plan for this project's Reproducible Research;

Version Control System: We will use Git to track changes to the code and data. This will make it easy to reproduce the project results at any point. We have already used git for this. The repository link is: https://github.com/Turjo7/WQD-7001-Group-Project-Data-Product

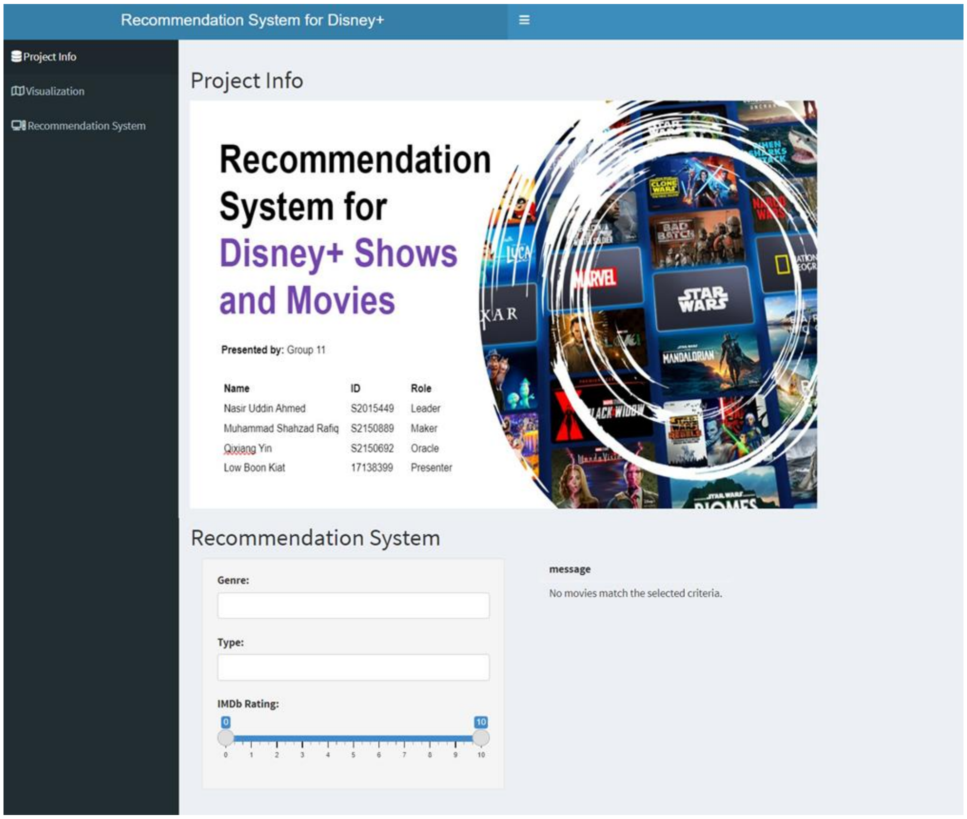
Document the Code: We will document the code with comments that explain what each part of the code does. This will make it easier for others to understand and modify the code. We will use a consistent coding style and will follow industry best practices for writing code.

Data Repository: We will store the data in a data repository, such as a Google Cloud Storage bucket or AWS. This will make it easy to share the data with others. We will use a consistent naming convention for the data files and will provide detailed documentation about the data.

Reproducibility Validation: Validate the reproducibility of the research by sharing the documentation, code, and dataset with others. Provide instructions on how to reproduce the entire project, including data acquisition, preprocessing, model training, and result interpretation. Encourage others to replicate the research and provide feedback or suggestions for improvement.

7. Deployment

The interactive web application was deployed on Shinyapps.io. This is a free service that allows us to deploy Shiny apps without having to set up our own server.



Project Info

Recommendation System for Disney+ Shows and Movies

Presented by: Group 11

Name	ID	Role
Nasir Uddin Ahmed	S2015449	Leader
Muhammad Shahzad Rafiq	S2150889	Maker
Qixiang Yin	S2150692	Oracle
Low Boon Kiat	17138399	Presenter

Recommendation System

Genre:

Type:

IMDb Rating:

message

No movies match the selected criteria.

Figure 4: Deployment of the Data Product

In our interactive web application, we require three inputs from users to generate personalized recommendations:

a) Genre: Users can select their preferred genre from a dropdown menu. This input helps narrow down the recommendations based on the user's interests.

b) Type: Users can choose between movies, series, episodes using a dropdown menu. This input allows users to specify their preference for the type of content they want to explore.

c) IMDb Rating: Users can input a minimum IMDb rating using a slider. This input allows users to set a threshold for the minimum rating they expect from the recommended shows or movies.

Overall, our web application aims to provide an enjoyable and personalized experience for Disney+ users, allowing them to discover new shows and movies tailored to their preferences.

8. Future Work & Conclusion

Expanding the dataset will allow us to capture a broader range of user preferences and interactions, leading to more accurate and diverse recommendations. A larger dataset provides a richer source of information, enabling us to extract meaningful patterns and insights to deliver high-quality recommendations to users.

Incorporating more sophisticated algorithms enhances the recommendation engine's capabilities. Advanced techniques such as collaborative filtering, content-based filtering, or deep learning models can be employed to capture intricate relationships and generate highly personalized recommendations. These algorithms can learn from user behavior and preferences to provide more relevant and tailored suggestions.

Automation of the system streamlines the recommendation process, eliminating the need for manual intervention. By automating data collection, preprocessing, model training, and recommendation generation, the system can continuously update and adapt to user preferences. This ensures that the recommendations stay up to date and reflect the evolving interests of the users.

In conclusion, by upgrading our system with a larger dataset, utilizing more sophisticated algorithms, automating the system, and applying a hybrid approach, we can enhance the personalized recommendation web application for Disney+ shows and movies on R Shiny.

Overall, the personalized recommendation web application serves as a valuable tool for Disney+ users, enabling them to discover new shows and movies aligned with their interests. The intuitive user interface, coupled with the incorporation of advanced features and algorithms, enhances the overall user experience. As technology and data science continue to evolve, there is ample potential for further improvements and innovations in the realm of personalized recommendations, providing users with an even more tailored and enjoyable content exploration experience.

9. References & Appendixes – Slide Presentation, Modelling, User Manual etc.

Ahuja, R., Solanki, A., & Nayyar, A. (2019). Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor. 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 263–268. https://doi:10.1109/CONFLUENCE.2019.8776969

Awan, M. J., Khan, R. A., Nobanee, H., Yasin, A., Anwar, S. M., Naseem, U., & Singh, V. P. (2021). A Recommendation Engine for Predicting Movie Ratings Using a Big Data Approach. Electronics, 10(10), 1215. https://doi.org/10.3390/electronics10101215

Fortune Business Insights (2022) Video Streaming Market Size Forecast 2022-2029, Video Streaming Market Size, Share, Growth & Forecast [2029]. Available at: https://www.fortunebusinessinsights.com/video-streaming-market-103057 (Accessed: 08 May 2023).

Gomez-Uribe, C. A., & Hunt, N. (2015). The Netflix Recommender System. ACM Transactions on Management Information Systems, 6(4), 1–19. https://doi.org/10.1145/2843948

Richter, F. (2021) Infographic: Where Americans get their stream on, Statista Infographics. Available at: https://www.statista.com/chart/25382/most-used-video-streaming-platforms/ (Accessed: 08 May 2023).

Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. Journal of Big Data, 9(1), 59. https://doi:10.1186/s40537-022-00592-5

Schwartz, B. (2016) The paradox of choice why more is less. New York: Ecco.

Zhang, J., Wang, Y., Yuan, Z., & Jin, Q. (2020). Personalized real-time movie recommendation system: Practical prototype and evaluation. Tsinghua Science and Technology, 25(2), 180-191. https://doi:10.26599/TST.2018.9010118