**Capstone Project Update 2**

**Introduction**

Steck, H., Baltrunas, L., Elahi, E., Liang, D., Raimond, Y., & Basilico, J. (2021). Deep Learning for Recommender Systems: A Netflix Case Study. AI Magazine, 42(3), 7-18. https://doi.org/10.1609/aimag.v42i3.18140

This paper talks about the impact deep learning have when used for machine learning and recommendation systems. It uses Netflix’s famous competence in that field to better communicate its effectiveness. The paper begins by reviewing various recommendation tasks on Netflix and in their case study they find that different algorithms excel at different tasks. Deep Learning was believed not to have a significant improvement in performance when compared to highly tuned non-deep learning methods, however it was found to have significant advantages when dealing with a diverse and heterogeneous set of attributes in a dataset.

When first implementing a deep learning approach, there may be some difficulties in actually building the engine, however there are existing toolboxes that make it very easy to develop and provide flexibility to tune your model for what task is needed. When using a “bag-of-items” model, the team noticed that these models ignored temporal information. They then tried to move onto sequential models and learned that both these models were viable solutions for various tasks.

Their goal for their recommendation system was to ensure the best long-term satisfaction for their members. This is important to help make customers feel like they should continue renewal of their membership to continue watching more content. They mention that some of the challenges of this is that everyone individual is unique in term of their interest, taste, and context. One of the problems they encountered in development was that offline performance (which is driven by historical data) was not the same as online performance. Deep-Learning models also struggled with short-term actions as these could have noisy data, which can cause changes in the training objectives. These small changes in the training-objective would sometimes lead to big changes in the recommendations. Deep Learning models also may work for the majority of users but may fail for the minority.

Deep Learning tools are abundant and powerful, which is another benefit to it’s use. Tools such as TensorFlow, Keras, or PyTorch can be used to address recommendation problems. The use of multiple customized models working in unison in a greater ecosystem at Netflix created an unparalleled level of efficiency as they noted. Applying deep learning technologies can help solve problems that traditional methods struggled with such as images, text, and videos and their various modalities. Deep Learning can also amplify weaknesses in recommendation systems which will help developers better tune their models when addressing those issues.

Ko, H., Lee, S., Bocanegra, S., & Choi, A. (2022, January 3). A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. MDPI. Retrieved October 8, 2023, from https://www.mdpi.com/2079-9292/11/1/141.

This paper examines how and which advanced recommendation systems trended between 2010 and 2021, and what were the business aspects of those services. They used papers and articles published on Google scholar to investigate the topic. They then took this data, systemized them as well as analyzed research trends by year. This paper also discusses diving into each of these recommendation systems and classified the application service fields the systems would be used for as well as the techniques used were analyzed. They aimed to do this because they note that the spread of the internet, smart devices, and Social Network Services have led to an increase in number and development of web and application-based services. These services often rely on various recommendation systems to receive data then make or assist in the decision-making process of the user. As more and more data is introduced, this process needs to make decisions quicker and recommendation systems have improved to combat this problem.

The paper found a correlation between the increase in research/papers published of recommendation systems, and the business growth in the applied fields related to the recommendation system. The researchers note that recommendation systems that utilize real-time data were able to drive better outcomes oftentimes. For example, wearable data that constantly gathered information helped aid in better diagnostic and treatment outcomes for patients even if clinical data provided more accurate information. It is important to note that the data here did not work alone, but instead it was used as supplemental to clinical data and provided real-time data.

The recommendation systems studied generally had 2 parts where there was a data mining segment as well as a recommendation filtering model. Each one of these systems were highly customized to perform the best at whatever task it was assigned to. In terms of model usage, the Content-Based filtering model has moved to be used alone. Collaborative Filtering models meanwhile seemed to be used in a hybrid model to complement it’s strength and weaknesses.

When looking into the filtering models of recommendation systems, text filtering was a method that had been used and studied over an extended period of time. However, Neural Network interest has increased recently, as more organizations and individuals look to implement this system.

This article also noted that real world companies such as Netflix, Amazon, and Yahoo are actively using these various systems in their recommendation algorithms according to papers published by representatives of their companies.

**Methods**

The data for this project will be taken from the online data community, Kaggle. Shivam Bamsal, a user in the Kaggle community, provided 4 separate datasets with data from Amazon Prime, Netflix, Disney Plus, and Hulu [4, 5, 6, 7]. These datasets has everything someone might want to use in terms of associating different shows such as release date, title, cast, and most importantly: category tags. I have adjusted the IDs of each piece of content to ensure no overlap then append all the data into a single sheet before adding the file to the data frame. Since all the category tags and cast tags are all split based on the delimiter “,”, I will most likely need to split the data based on that, so my algorithm can read all the various tags separately.

The description field in the data will also need to be disseminated to find common themes between content. This is a very important step to help the model find related content. To do this I will need to implement a form of NLP (Natural Language Processing). According to at least 1 paper, a good method to approach this would be using RAKE (Rapid Automatic Keyword Extraction). They note that based on their benchmarking, RAKE was found to be “more computationally efﬁcient than TextRank while achieving higher precision and comparable recall scores” [14]. I then found a repository in Kaggle that attempt a similar type of problem I was facing and used part of their code to help attempt to perform keyword extraction [15].

Scraping and/or distributing data directly from Netflix, Disney Plus, and Hulu are expressly denied based on the terms and conditions of these various platforms [11 , 12, 13]. Because of this I decided to use an already public database to ensure no legal repercussions regarding this data. None of the data used has personal information or information not available to the public.

I created a GitHub Repository to host my code as well as other project deliverables such as this project update. The public repository can be found in the following link: https://github.com/RafeedU/Unified-Content-Recommendation-System

**References**

[1] Cranz, A., & Becker, D. (2022, December 14). The golden age of the streaming wars has ended. The Verge. Retrieved October 8, 2023, from https://www.theverge.com/2022/12/14/23507793/streaming-wars-hbo-max-netflix-ads-residuals-warrior-nune

[2] Netflix. (n.d.). How Netflix's Recommendations System Works. Netflix Help Center. Retrieved October 8, 2023, from https://help.netflix.com/en/node/100639

[3] Ko, H., Lee, S., Bocanegra, S., & Choi, A. (2022, January 3). A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. MDPI. Retrieved October 8, 2023, from https://www.mdpi.com/2079-9292/11/1/141.

[4] Bansal, S. (2021, October 18). Netflix Movies and TV Shows. Kaggle. Retrieved October 8, 2023, from https://www.kaggle.com/datasets/shivamb/netflix-shows

[5] Bansal, S. (n.d.). Amazon Prime Movies and TV Shows. Kaggle. Retrieved October 8, 2023, from https://www.kaggle.com/datasets/shivamb/amazon-prime-movies-and-tv-shows

[6] Bansal, S. (n.d.). Disney+ Movies and TV Shows. Kaggle. Retrieved October 8, 2023, from https://www.kaggle.com/datasets/shivamb/disney-movies-and-tv-shows

[7] Bansal, S. (n.d.). Hulu Movies and TV Shows. Kaggle. Retrieved October 8, 2023, from https://www.kaggle.com/datasets/shivamb/hulu-movies-and-tv-shows

[8] Steck, H., Baltrunas, L., Elahi, E., Liang, D., Raimond, Y., & Basilico, J. (2021). Deep Learning for Recommender Systems: A Netflix Case Study. AI Magazine, 42(3), 7-18. https://doi.org/10.1609/aimag.v42i3.18140

[9] Canal, A. (2019, March 9). Your streaming bill is about to go up even more. yahoo!finance. Retrieved October 8, 2023, from https://finance.yahoo.com/news/your-streaming-bill-is-about-to-go-up-even-more-200938833.html?guccounter=1&guce\_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce\_referrer\_sig=AQAAAFYZ9uhxknmK6RJAOHi33nMu9PyDzuE7Vk3vnrV2rkhtDggUk-dZ-aBzdu29m5zbAurH8vAdo\_ZTQ0O80

[10] Ilmi, M. F. (2022, October 27). Hybrid and Tensorflow Recommender System. Kaggle. Retrieved October 8, 2023, from https://www.kaggle.com/code/mfaaris/hybrid-and-tensorflow-recommender-system

[11] Disney+. (2022, September 27). Subscriber Agreement. Legal. Retrieved October 22, 2023, from https://www.disneyplus.com/legal/subscriber-agreement

[12] Hulu. (2022, October 24). Hulu Subscriber Agreement. Hulu. Retrieved October 22, 2023, from https://www.hulu.com/subscriber\_agreement

[13] Netflix Queue. (2023, January 25). Privacy Statement and Terms of Use. Netflix Queue. Retrieved October 22, 2023, from https://netflixqueue.com/privacy

[14] Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic Keyword Extraction from Individual Documents. In Text Mining: Applications and Theory (pp. 1-20). Wiley. 10.1002/9780470689646.ch1

[15] Singh, V. (2019, March 9). Netflix Movies and Shows: Plotly & Recommender SYS. Kaggle. Retrieved October 22, 2023, from https://www.kaggle.com/code/vikassingh1996/netflix-movies-and-shows-plotly-recommender-sys/notebook

**Appendix A: Bard**

1. If I have a dataframe in Python and the first column is the ID of my content. Each ID has 1 letter (either d, p, n, or h) then a series of numbers. I want to add a column in my dataframe that uses that ID column as an input so that the new column will label each ID as Disney for D, Prime for P, Netflix for N, and Hulu for H.

A screenshot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

1. Is using a combination of RAKE and NLTK the best way to perform Keyword Extraction in a dataframe?

A black screen with white text

Description automatically generated

A black screen with white text

Description automatically generated

1. Please make me a table listing strengths and weaknesses between KNN, Support Vector Machines, and Neural Network Machine Learning Models for a TV show recommendation System that uses primarily qualitative features in the dataset

A screenshot of a computer

Description automatically generated

A black screen with white text

Description automatically generated