**A Movie Recommender Engine:**

**A Content-Based Approach to Recommend International Movies**

**David Thor**

[**DavidInData@gmail.com**](mailto:DavidInData@gmail.com)

[**https://github.com/DavidInData/movie\_recommender**](https://github.com/DavidInData/movie_recommender)

<https://recommender15-mzj2jqef7a-uc.a.run.app/>

**November 2022**

**Abstract**

There are currently no reliable recommender engines for international films as most online streaming platforms and movie recommenders are based in the United States. Recommender engines today rely heavily on western-based movie review sources such as IMDb, Rotten Tomatoes, or Metacritic. Using content-based filtering and machine learning techniques, this research presents a movie recommendation engine on international movies based on reviews from MyDramaList, a very popular international movie review website within the Asian community. The deliverable will be a interactive web application hosted on Google Cloud Platform.

Keywords: *International Film, Recommendation Engine, Content-Based Filtering*

**Introduction**

With the rise of online movie streaming and robust recommendation engines, there is still an underrepresented recommendation for international movies. The recommendation engine algorithm has been increasingly popular in the last decade and is the sole competitive advantage to some of top tech companies. In 2016, Business Insider related an article titled, “Why Netflix thinks its personalized recommendation engine is worth $1 billion per year.” Netflix’s Chief Product Officer Neil Hunt, they assert that “the combined effect of personalization and recommendations save us more than $1B per year." (McAlone, 2016). One cannot deny the successes of the recommendation engines. However, some users are still experiencing poor Asian movie recommendations due to the lack of data. Recommender engines today rely heavily on western-based movie review sources such as IMDb, Rotten Tomatoes, or Metacritic. For comparison, a movie like *Harry Potter and the Prisoner of Azkaban* has 148 entries on the IMDb trivia section compared to one of the most well-known Asian movies *Crouching Tiger, Hidden Dragon* which only has 32 entries. Even a film like *Akira*, one of Japan's most well-known anime films has only 46 entries.

Recommendation improvements will be a continuous process. For instance, Netflix finally released its first “global” recommendation engine in December 2016 (McAlone, 2016). Shortly after, Netflix’s success with ‘Squid Game’ is making other streamers eager to explore international production (Sherman, 2021). Although there have been improvements the recommender engines in the recent years, the underlying driving factor to address the poor recommendations are the data sources. Specifically, recommendation engines must rely on quality international content to drive their recommendations. Using content-based filtering and machine learning techniques, this research presents a movie recommendation engine on international movies based on reviews from MyDramaList, a very popular international movie review website within the Asian community.

**Literature Review**

The concept behind a recommender system is to suggest to users an item or product based on some given data. In this study, the objective is to recommend a list of similar movies based upon an initial movie selection. There are many approaches to a recommender system and can be classified into three categories; collaborative-filtering, content-based, or a hybrid between the two. Collaborative recommender systems aggregate ratings or recommendations of objects, recognize commonalities between the users on the basis of their ratings, and generate new recommendations based on inter-user comparisons. They work well for complex objects where variations in taste are responsible for much of the variation in preferences. Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future and that they will like similar kind of objects as they liked in the past (Belhekar, 2020). A content-based recommender learns a profile of the new user’s interests based on the features present, in objects the user has rated. It’s basically a keyword specific recommender system here keywords are used to describe the items. Thus, in a content-based recommender system the algorithms used are such that it recommends similar items that the user has liked in the past or is examining currently (Belhekar, 2020). Hybrid recommendation algorithm is the organic fusion of multiple different single recommendation algorithms. As a combination strengthening algorithm, it can not only effectively avoid the shortcomings of single algorithm but also enhance the recommendation efficiency of hybrid algorithm, thus improving the recommendation performance (Miao, 2022).

Diagram

Description automatically generated

Figure 1: Collaborative & Content-based Filtering © Belhekar, 2020

In the recent years, there has also been popularity with another deep-learning based recommendation algorithm. Autoencoder is a special kind of neural network, which can learn the deep hidden features of input data by adding constraints to the model. The autoencoder can compress and reduce the dimension of data, learn the deep features of the high-dimensional sparse data, and compress and generate the low-dimensional hidden feature vectors in the hidden layer. Encoder neural network can not only reduce and compress the high-dimensional data but also learn the deep features of the target, so it is often used to extract the hidden features of the target (Miao, 2022). To provide a bird-eye’s view of this field, we classify the existing models based the types of employed deep learning techniques. We further divide deep learning based recommendation models into the following two categories shown in Figure 2 (Zhang, 2019).

Diagram

Description automatically generated

Figure 2:Categories of deep neural network based recommendation models. ©Zhang, 2019

There are currently no other studies conducted for this given dataset. The most notable recommender engines are collaborative filtering, and their respective datasets are specific to their own streaming platforms such as Netflix, Hulu, Amazon Video, etc. As for independent recommender systems, they utilize movie datasets heavily biased upon western content or contain abundance of reviews for accurate recommendations. To address this, this research presents a unique content-based recommendation study utilizing machine learning techniques on a specific dataset with reliable features.

**Data Source**

The data source for this recommender engine is from MyDramaList.com. MyDramaList is a community-based project which provides Asian drama & movie fans a site where they can rate dramas and films, write reviews, and make recommendations. MyDramaList launched on April 6th 2011, and currently has approximately 500K supporters. This data source was selected due to its popularity within the international community. Other film critics have referenced it as the IMDb for international/Asian films.

This dataset for this study consists of movie reviews on MyDramaList and the data was scraped on February 2022 using the BeautifulSoup Python Library. The size of the dataset is 9657 Asian Movies and Drama including its features. The features for each movie consist of the title, type, country, director, synopsis, episodes, score, aired, duration, genres, tags, and cast names. As for data preparation, one feature was engineered utilization several variables and string concatenation across director, genres, tags, synopsis and cast names. These variables were selected as the contributing features because of their reliability and dependability for movie identification. Below is a histogram representing the content’s country of origin across the dataset.

Chart, bar chart

Description automatically generated

Figure 3: Most Movies/Drama based on Country of Origin

**Methods**

Figure 4: Data Preparation, Pre-processing, TF-IDF, Cosine Similarity

Diagram

Description automatically generated

The recommendation approach for this study is content based, similarities in products, services, or content features, as well as information obtained about the consumer, are used to produce suggestions (Joseph, 2022). See Figure 4 for visual representation of the process including data pre-processing and the respective methods.

The first step in the process is the data preparation discussed above. This step involved feature engineering, removing null values, assuring each word is properly spaced in the data. The pre-processing in step two involves utilizing TfidfVectorizer to eliminate stop words. Removing English stop words are necessary as it provides no meaning for any sentiment analysis. Thirdly, TF-IDF vectorization is generated. Sklearn’s TfidfVectorizer is used to vectorize and calculate the TF-IDF for each word within features in the corpus of movies. Term Frequency-Inverse Document Frequency (TF-IDF) is a method to determine importance of a term in the document as it relates to the corpus (Garla, 2021). In short, the TF-IDF Matrix provides meaning to each word in a particular movie as it relates to all the movies.

Text

Description automatically generated

Figure 5 - TF-IDF: A Visual Explainer © Anupama Garla

After the TF-IDF matrix, that is, we understand the importance of each movie in terms of words, we can generate a Cosine Similarity Matrix to determine ‘like-movies’. Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis (Han, 2011). In other words, the Cosine Similarity Matrix will allow us to determine which movie to recommend based upon the cosine similarity score. Figure 6 and 6.1 provide visual representations of the Cosine Similarity Scores. Similar vectors (movies) will yield a cosine similarity score closer to 1. Notice how Avengers & Thor have a higher similarity score compared to Avengers & Parasite. This is because Avengers & Thor are both similar in terms of plot, cast and genre. The final step in the process is user interface. A user interface will be created for ease of accessibility. A list of ten movies will be recommended upon selecting an initial movie.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cosine Similarity Scores** | **Avengers** | **Thor** | **Parasite** | **Karate Kid** |
| **Avengers** | 1 | 0.8 | 0.3 | 0.6 |
| **Thor** | 0.8 | 1 | 0.4 | 0.75 |
| **Parasite** | 0.3 | 0.4 | 1 | 0.2 |
| **Karate Kid** | 0.6 | 0.75 | 0.2 | 1 |

Figure 6: Cosine Similarity Matrix

**Diagram

Description automatically generated**

Figure 6.1: A Cosine Similarity Visual © Karabiber

**Text

Description automatically generated**

Figure 6: Cosine Similarity © Han, 2021

**Implementation**

The implementation to Google Cloud Platform consists of many parts. First off, the development began on my local machine using Python and Flask, a web application framework. However, GitHub was the primary central repository for the source data. All the source code were pushed to GitHub after successful testing locally. Afterwards, the source code was cloned to Google Cloud Platform’s Cloud Shell via Git commands. In addition, there were triggers built in Google Cloud to read the movie data, create a Google Cloud Bucket, and made available in BigQuery. The purpose here was to enable additional reporting capabilities and further analysis for super users.

Google Cloud Run was decided as the service provider due to several reasons as opposed to App Engine or Compute Engine. Cloud Run’s pricing were significantly cheaper compared to App Engine or Compute Engine. Specifically, Cloud Run costs 99% less than App Engine and if you’re a hobbyist developer and you want to host your fun app for next-to-free, you should definitely use Google Cloud Run (Karabiber, 2020). As for comparison with other cloud providers. There are two main advantages Google has compared to Azure or AWS. First, the visibility on pricing. Google provides an extremely transparent pricing model for any service. Secondly, the Stackdriver monitoring service unique to only Google. As seen in the comparison below by ISmile Technologies, Stack Driver provides logging, error reporting, debugger and more. Many of these can be advantageous not only to technical individuals but also business users.

Timeline

Description automatically generated

Because of this unique use-case, a custom algorithm was determined and a decision was made not to leverage AutoML or extensive BigQuery. We can first look at how TCO can be lower with AutoML. AutoML in general works well with non-tech savvy individuals or firms that may not be able to invest in a Data Scientist. AutoML would work perfectly as it's less technical and everything is self-managed. The individual/firm would have very little 'cost' to maintain or support the model. The pros definitely beats the cons in this scenario as I'd imagine start ups won't do any extravagant complicated models. On the other hand, AutoML can have a higher TCO for more complex algorithms that is built to meet the business requirement. For example, a fortune 100 tech firm would have to invest a lot of money in AutoML to train/support models for business needs assuming they do not have any custom trained models or internal Data Scientists. Many peers have implemented analytical models in differing approaches. For instance, we've seen model deployments using BigQuery and AutoML into App Engine or implementing to completely to AWS. Others have deployed with a really refined ETL process integrating full circle CI. Others have implemented sophisticated deep learning models. There's always a balance and the important thing is to understand the business requirement and solution something with our toolset/tech to meet the needs.

**Directions for Future Work**

There are certain limitations to content-based filtering. For example, movie recommendations are limited to the movie’s meta-data. Moreover, there are many improvements for the implementation to Google Cloud. First, continuous improvement and continuous deployment. Currently, it is a manual process to push a new version of the app to Cloud Run. Secondly, integration with BigQuery or a cloud database can significantly improve performance and consistency. This would also align to industry best practices. Lastly, automation with scraping the movie data. This was a one-time manual process and would need to be scraped once more in the near future. Automating this process all the way through (end-to-end) would be the ideal state.

**Conclusion**

Using content-based filtering, this research presented a movie recommendation engine on international movies based on reviews from MyDramaList, a very popular international movie review website within the Asian community. Utilizing TF-IDF and Cosine Similarity Matrices, the core machine learning techniques for this study, recommendations were noticeably different compared other recommenders such as Google and Rotten Tomatoes. The model was also deployed to Google Cloud Run with many opportunities for improvements as discussed.

**Bibliography**

“Amazon Cloudwatch vs Google Stackdriver vs Microsoft Azure Monitor.” ISmile Technologies, October 25, 2021. https://www.ismiletechnologies.com/technology/amazon-cloudwatch-vs-google-stackdriver-vs-microsoft-azure-monitor/.

Belhekar, Chaitanya. “Recommendation Systems - a Walk Through.” Medium. Medium, August 28, 2020. <https://chaitanyabelhekar.medium.com/recommendation-systems-a-walk-trough-33587fecc195>.

Johansen, D. R. (2020, October 12). Automl - is it useful? - automatic machine learning. Dan Rose AI. Retrieved October 25, 2022, from https://www.danrose.ai/blog/automl-is-it-useful

Karabiber, Fatih. “Cosine Similarity.” Learn Data Science - Tutorials, Books, Courses, and More. Accessed November 21, 2022. https://www.learndatasci.com/glossary/cosine-similarity/.

Garla, Anupama. “TF-IDF : A Visual Explainer and Python Implementation on Presidential Inauguration Speeches.” Medium. Towards Data Science, February 8, 2021. https://towardsdatascience.com/tf-idf-a-visual-explainer-and-python-implementation-on-presidential-inauguration-speeches-2a7671168550.

Han, Jiawei, Micheline Kamber, and Jian Pei. “Getting to Know Your Data.” Data Mining (Third Edition). Morgan Kaufmann, June 24, 2011. https://www.sciencedirect.com/science/article/pii/B9780123814791000022?via%3Dihub.

Joseph, Aleena, and Ms. Jetty Benjamin. “Movie Recommendation System Using Content-Based Filtering and Cosine Similarity.” Zenodo, July 2, 2022. <https://doi.org/10.5281/zenodo.6791117>.

Karabiber, Fatih. “Cosine Similarity.” Learn Data Science - Tutorials, Books, Courses, and More. Accessed November 21, 2022. https://www.learndatasci.com/glossary/cosine-similarity/.

McAlone, Nathan. “Why Netflix Thinks Its Personalized Recommendation Engine Is Worth $1 Billion per Year.” Business Insider. Business Insider, July 14, 2016. <https://www.businessinsider.com/netflix-recommendation-engine-worth-1-billion-per-year-2016-6>.

Miao, Guojian, Yin Gao, and Zhenshen Zhu. “Digital Movie Recommendation Algorithm Based on Big Data Platform.” Mathematical Problems in Engineering. Hindawi, May 31, 2022. https://doi.org/10.1155/2022/4163426.

Reddy, SRS, Sravani Nalluri, Subramanyam Kunisetti, S. Ashok, and B. Venkatesh. “Content-Based Movie Recommendation System Using Genre Correlation.” SpringerLink. Springer Singapore, January 1, 1970. https://link.springer.com/chapter/10.1007/978-981-13-1927-3\_42.

Sherman, Alex. “'Squid Game' Success Shines a Light on How Cheap It Is to Make TV Shows Outside the U.S.” CNBC. CNBC, October 16, 2021. https://www.cnbc.com/2021/10/16/netflixs-squid-game-success-shines-light-on-international-discounts.html.

Zhang, Shuai, Lina Yao, Aixin Sun, and Yi Tay. “Deep Learning Based Recommender System: A Survey and New Perspectives.” arXiv.org, July 10, 2019. <https://doi.org/10.48550/arXiv.1707.07435>.