Movie Recommendations

A Look at Recommendation Systems and the Netflix Dataset

Abdullah Adlouni
Applied Computer Science
University of Colorado
Boulder, CO
abad2907@colorado.edu

Charles Dorman
Applied Computer Science
University of Colorado
Boulder, CO
chdo1939@colorado.edu

Ben Healy
Applied Computer Science
University of Colorado
Boulder, CO
healybg@colorado.edu

David Shackelford
Applied Computer Science
University of Colorado
Boulder, CO
dash9231@colorado.edu

Problem Statement/Motivation

Video streaming services are a large and growing market in the entertainment industry, and the growing competition in this space is more important than ever. The streaming industry is worth an estimated $60.1 Billion in 2021 and is estimated to be worth over $330 Billion by 2030.[1][2] In addition to this, streaming services have a higher amount of traffic than ever before. 86% of households own at least one video streaming service in the US, with streaming services being accessed by over 110 households in 2021.[3] This large amount of growth is partly attributed to the increasing use of big data and integrating information and statistical analysis with the goal to provide a better service to customers. With more competition than ever, it also becomes increasingly important for a company to use this large amount of data to improve their product. With so many different services offered, it is imperative to keep people watching and engaged instead of moving to a better service. This can be done through data as it can highlight hidden insights within a customer base and provide recommendations to the user on what to watch next, or to provide the company with information to know what types of content are better to invest time and money.

The main purpose of this project is to explore the data that large streaming services use in order to gain some insight into what movies people enjoy watching. To accomplish this, we have 3 main goals. The first goal is to create recommendations based on prior knowledge of customer ratings. To accomplish this goal, we will be looking at a dataset provided by Netflix for competitions. Doing this will allow us to look at recommendation systems further in the context of real data and to provide a baseline for our analysis. The second goal we have is to use other attribute data available to create additional inferences and insights. This will entail getting other resources such as data taken from IMDB. The third goal is to observe trends within the movie industry. This involves taking our conclusions from our previous work and adding better explanations of the movie industry through visualization and further analysis.

KEYWORDS

Big Data, Data Mining, Machine Learning, Movie Reviews, NLP, Recommendation

Reference

Abdullah Adlouni, Charles Dorman, Ben Healy, and David Shackelford. 2022. Movie Recommendations: A Look at Recommendation Systems and the Netflix Dataset.

1 Literature Survey

As mentioned, streaming services are a large and growing industry. As such, there have been many research papers on recommendation systems for movies. One of the biggest pieces of work related to this project is known as the Netflix prize,[7] a $1 million competition issued in 2009, which challenges data scientists to implement a collaborative filtering algorithm using real collected data. The top algorithms in this competition are still used today. The winner of this competition was the Bellkor's Pragmatic Chaos Team, which created a recommendation system that was 10% better than Netflix's existing algorithms by using ensembled learning to create a better recommendation system.[8] The Netflix dataset we are using in this project is the same one used for the competition. Using this work as an example will help us navigate through the project.

Another important piece of work is a paper on improving Netflix data based on outside resources.[9] In this, Bhatia *et al* try to use outside resources such as the IMDB dataset in order to improve on the Netflix prediction score. They chose 3 main attributes to improve their results – Genre, Director, and Actor – which when combined with collaborative Netflix movie ratings gives an RMSE for recommendation of 0.915 which is better than the model error when not including added information (0.867). The team mentioned that they were unable to apply added techniques such as NLP to match titles, and that their implementation could greatly improve without the time constraints.

The changes that we will observe in our project will be two-fold. First, unlike the Netflix prize results, we will utilize multiple datasets in order to improve our recommendation model as much as possible. Second, on top of using other data sources, we will examine additional integration methods, such as NLP, and will also try multiple models in order to maximize our results. We will also utilize data visualization to try and get more useful information from our combined datasets.

2 Proposed Work

Work will be split into 5 parts: data collection, preprocessing, integration, design, and evaluation.

2.1 Data Collection

Each dataset is collected online and uploaded to a public AWS S3 bucket (moviereview.data). This is to ensure that everyone has access to the same datasets. Some datasets are very large and so it would be inefficient to upload and access all datasets through the github repository.

2.2 Preprocessing

Preprocessing will involve removing any blank values, duplicated values, or wrong values. This includes any custom NULL values or values that don't make sense. As part of the integration step, we also need to remove values which don't contain a matching pair for the other datasets. This is found during integration and should be kept to a minimum in order to retain the dataset's integrity.

2.3 Data Integration

Our team then needs to combine the Netflix, IMDB, and TMDB datasets together by using the title name attribute as a key pair. There are a few options to implement this. One is to purely match the titles. This may be the easiest but will produce the worst results. Another option is to use NLP to vectorize the titles and match them based on cosine similarity. This is reliant on a vectorization model such as Google's word2vec model and will require extra preprocessing such as stemming or lemmatization. Another option is to use an algorithm to match title close-ness. Each method will be tried and compared to get the best results.

2.4 Design

After integration, we will implement a recommendation model. For this, we will leverage multiple modeling methodologies to provide the most stable and consistent model results. We will first create a base model by only using Netflix data. We will then apply other data taken from outside sources to try and improve on the model's score. Since there are multiple methods for building a recommendation system, each team member will be responsible for creating a model and the results of all models will be examined after.

2.5 Evaluation

A summary of model performance will be provided in the final writeup that will discuss the reason for final model selection given a specific prompt. The models will be compared based on RMSE.

3 Dataset

Our project makes use of the Netflix Prize dataset, the IMDb datasets, and the TMDB dataset.

3.1 Netflix

The Netflix dataset is taken from Kaggle (https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data) and is created by Netflix for the use of the Netflix Prize competition.[7] The Netflix Prize dataset is made up of three subsets of data, the training dataset, the probe dataset, and a movie file with movie info. The training dataset contains over 100,000,000 movie ratings from 480,189 users on 17,770 different movies.[4] The data is in the format MovieID: user,

rating, date of rating. The movie file contains movie information in the format MovieID, YearOfRelease, and Title. The probe dataset with ratings can be used to train models by reducing the RMSE of predicted ratings against actual ratings.

3.2 IMDB

The IMDB dataset is taken from IMDB's main website (https://datasets.imdbws.com/) and is a very large set of datasets detailing most movies, tv shows, and specials that have been released throughout history. The IMDB datasets have information on 7,980,307 unique movies, shows, and shorts and 11,906,873 unique cast members.[5] The data sets are broken down into 7 subsets:

1. Title.akas - has regional title name, language, region, and type, with 8 attributes in total.
2. Title.basics – has secondary name, year released, runtime, genre, with 9 attributes total.
3. Title.crew – has info on directors, actors, with 3 attributes total
4. Title.episode - has info on show titles, and episode info with 4 attributes total.
5. Title.principles - has info on the principle cast and crew including ordering and specific role with 6 attributes total.
6. Title.ratings - has IMDB rating and vote information with 3 attributes total.
7. Title.name.basics - has crew information including names, date of birth, date of death, and known-for-titles, with 6 attributes total.

3.1 TMDB

The TMDB datasets are a tertiary dataset which consist of added information from the IMDB datasets. This set of data was collected from Kaggle's website (https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata). The TMDB dataset consists of two files with info on 4,800 movies.[6] The credits file contains information in the format columns: movie id, title, cast, crew. The movies file contains information in the format columns: budget, genres, homepage, id, keywords, original language, overview, popularity, production companies, production countries, release date, revenue, runtime, spoken languages, status, tagline, title, vote average, vote count.

4 Evaluation Methods

Evaluation will be used for both data integration and model selection/performance.

4.1 Integration

To test that we were able to successfully integrate both datasets, we will have two metrics. The first metric is the amount of lost data. If we are not able to find a key pair for the datasets, we will drop that object as it won't be useable. Therefore, we want to minimize number of objects that we remove. The second metric is accuracy. To do this, we will check other attributes for the object such as release year and director name and see if the values match. We will use the integration method with the highest accuracy and the lowest data loss.

4.2 Data Model

The goal of our data model is to maximize both accuracy and generalization. For this, we will try multiple different models and determine the best choice based on classification metrics. This includes creating a confusion matrix which identifies rates of true-negative (TN), true-positive (TP), false-positive (FP), and false-negative (FN) between actual values and predicted values. Accuracy and RMSE will identify how often our class is correct:

$$accuracy = \frac{TP + TN}{N}$$

Precision will give us an idea of quality in our results:

$$precision = \frac{TP}{TP + FP}$$

Recall will show us our model's ability to find all relevant instances in a class:

$$recall = \frac{TP}{TP + FN}$$

The goal is to maximize the above metrics while also minimizing the RMSE.

5 Tools

Tools are broken up between project management, data analysis, and machine learning.

5.1 Project Management

The group meets in weekly scrum meetings on Thursdays for 30 minutes using Google Meet. Communication is maintained using email and discord. Code and documentation are accumulated using a github repository (github.com/dash2927/ABCD). Each team member works in their own branch with each branch being merged onto a main branch for major milestones and breakthroughs. To maintain consistency and access to data, AWS S3 will be utilized. AWS S3 is a simple storage service which will allow us to culminate datasets. A public bucket (moviereview.data) has been made to access all files. Boto3, a python library, will interface with AWS to retrieve data.

5.2 Data Analysis

Pandas and Numpy are used for major data analysis. These are python libraries which will help us manipulate csv data and perform calculations. The python library Seaborn will be utilized for data visualization. Regex (regular expression) is used to quickly search for text patterns to facilitate the merging of databases, specifically to overcome differences in movie titles between databases. Bash scripting may be used at a later point in the project for pipeline data manipulation, e.g., to assess the model across various metrics.

5.3 Machine Learning

For machine learning, the NLP toolkit in Python, NLTLK, has been useful in merging databases with differing titles for the same movie, and may be useful in determining worthwhile attributes to pull from the secondary databases being merged in. Python's ML library Scikit-Learn will be used for different models. The Python library Statsmodels may be used for vectorization during NLP. We may assess the use of a NN model via Tensorflow as well.

6 Milestones

Milestones for this project are based on a final submission of Dec. 8th.

6.1 Integration – Nov. 3rd

Find best method to integrate data sources.

6.2 Preprocessing – Nov. 10th

Full EDA should be performed on data. This includes cleaning and preprocessing data. Preliminary data visualizations should be finished in order to help build research story.

6.3 Model Building – Nov. 21st

Create models based on individual team member's subtasks. Generate model suggestions and compare to pick the best model.

6.4 Visualizations – Nov. 25th

Create visualizations that support findings.

6.5 Finalization – Dec. 5th

Piece together all analysis for final presentation and report.

REFERENCES

[1] https://www.grandviewresearch.com/press-release/global-video-streaming-market
[2] https://www.blueweaveconsulting.com/report/video-streaming-market
[3] https://www.kantar.com/north-america/inspiration/technology/us-video-streaming-market-growth-stalls
[4] https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data
[5] https://datasets.imdbws.com/
[6] https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata
[7] https://journals.sagepub.com/doi/full/10.1177/1461444814538646#bibr66-1461444814538646
[8] Jahrer, Michael & Töscher, Andreas & Legenstein, Robert. (2010). Combining predictions for accurate recommender systems. 693-702. 10.1145/1835804.1835893.
[9] Bhatia, N., & Patnaik, P. (2008). Netflix Recommendation based on IMDB.