Netflix Movie Reccomendation System

Gavin Galusha
Tulane University / CA
ggalusha@tulane.edu

Gabe Epstein
Tulane University / CO
gepstein1@tulane.edu

Abstract

Netflix Recomendation system, based on NLP and Machine Learning tools like word vectorization, collaborative filtering, cosine similarity, Content recomendation, Deep Neural Networks, and more.

1 Introduction

For our Project, we are building a Movie recomendation system. The project is built on an ensemble of methods: Firstly, Collaborative-Filtering filtering is a way to recommend movies to a user, based off the preferences of that user. In this case, this would be the movies this specific user has rated.

By aggregating the feature vector of each item (movie) a user has reviewed, we can build this user profile. From here we can use cosine similarity to compute the similarity between users. This already gives us a great start in predicting which movies a user may like, because odds are it is one that someone else who shares their movie taste also likes.

2 Related Work

Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms Outlines a method for enhancing Netflix's recommendation engine by employing TF-IDF and Cosine Similarity to analyze the textual content of Netflix's program offerings. This approach includes an exploratory data analysis (EDA) to understand trends in the type of programs offered, followed by the implementation of these NLP techniques to set up a recommendation system that evaluates program descriptions and user preferences for similarity, aiming to refine and personalize recommendations. This method provides a nuanced way to match content with user interests more accurately.

https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/18140

3 Methods

Mood Prediction:

Our first step in the overall process of movie generation, was to try to guess the mood of a movie based off it's other features. For this, we started by using CountVectorizer to create embeddings for the description sentences. We used one-hot encoding for the categorical features, and paired these with the embeddings generated by the descriptions as the input to predict the mood class for each film.

Next, we performed our train-test-split, and fit a logistic regression model for classification. Initial Accuracy is around 83 percent, but we're hoping to get this number up. By generating the mood for different films, we hope to add a further feature that can help us better create profiles for each user.

Cosin Similarity:

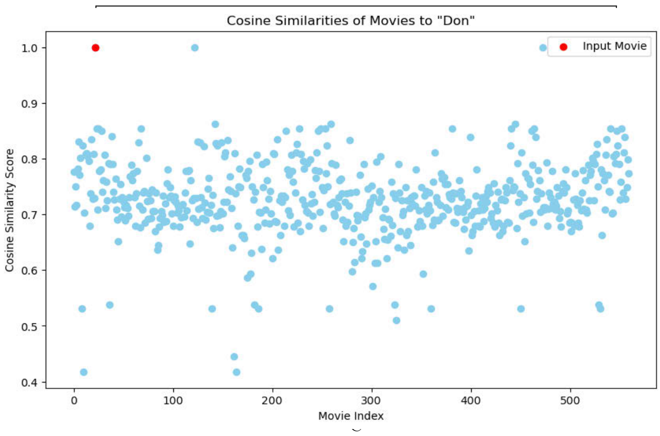
In experiments2, we took our first crack at cosine similarity. For this task, we sought to find the cosine similarity between movies by first encoding features, and then generating a similarity matrix with all movies in terms of eachother. This allows easy visualization of the movies. Initial output was succesfull, as seen when inputting the Movie "Don", the most similar movie was "Don 2".

Future Plans:

Our next step is to perform collaborative filtering on the user/movie data. We will create a User-Movie matrix. From here we can do a variety of techniques like more cosine similarity, matrix factorization, and more on this User-Movie Matrix.

4 Preliminary Results

The initial Results of cosign Similarity prove promising. The top blue dot next to the input dot is "Don 2", an easily recognizable similar film.



5 Data Preparation

There are two primary datasets that we have used for this project: https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset/code and https://www.kaggle.com/datasets/ashpalsingh1525/netflixLinks. The first dataset contains a variety of dataframes all linked together with keys like userId, movieId, and tagId. This one is essential for user-user collaborative filtering.

The second dataset contains specific movie attributes, like description, duration, cast, etc. This one is ideal for calculating movie-movie similarity. The first step was to join together the various dataframes, and string together a cohesive data frame with no missing values.

6 Conclusion

The evaluation of our recommendation system will rely on objective measurements derived from the system's performance on a held-out test dataset. We will use metrics such as precision, recall, F1 score, and Mean Average Precision (MAP) to quantify the system's accuracy in predicting user preferences based on their input and interaction history. Comparison with a baseline model, such as traditional collaborative filtering without real-time user input, will highlight improvements. We'll also utilize confusion matrices to analyze the model's performance in different recommendation scenarios, providing a detailed view of its strengths and areas for improvement. This method ensures an objective and comprehensive evaluation without the need for live deployment or user feedback.

7 Division of Labor

The work was divided between Gabe Epstein and Gavin Galusha on this project. Both members collaborated frequently, but Gavin focused more on the data gathering and pre-processing, while Gabe did more model evaluating and fitting