

NETFLIX

A Movie Recommendation AI

By Andrianto Reno (3035663473), Buntoro Reynard Verill (3035663461)



INTRODUCTION

Netflix is undoubtedly one of the leading companies in the tech-entertainment industry as they provide their customers with outstanding amounts of movies and tv-shows to watch on a subscription basis. Admittedly, their swift increase in popularity is also associated with their film recommendation algorithm, which allows their customers to choose their favorite films with ease. The previous statement is reinforced with the fact that about 80% of their streams resulted from their recommendation system (Chong, 2020). This recommendation system seemingly gives personalized suggestions for users at home to watch films with similar attributes which might suit their preference more. Therefore, in this project, we are trying to come up with a prediction model that is able to correctly predict what the users desire to watch with satisfactory accuracy using the dataset of netflix prize data, which was also utilized as the dataset for the 2006 Netflix Prize Open competition.

Project Objectives

This project aims to accomplish giving proper film recommendations towards Netflix viewers using a dataset provided by Netflix for their competition in 2005. Our recommendation system will try to give a list of personalized movies/shows suggestions based on past ratings that each user has given.

Application

The application for this project will be following the basic tutorial of data wrangling, cleansing, and others from an available project of the same dataset from kaggle <https://www.kaggle.com/laowingkin/netflix-movie-recommendation>, using similar approach to the one we are proposing as a mean of tutorial and reference. For the live, reactive application, we are considering making a Shiny dashboard application using R to allow users to interact with a reactive page for data visualizations, and ultimately for making predictions with some interactive inputs. Apart from this, we are also thinking of deploying our application using Flask as the backend with Python, along with a front-end (However we are still not sure of this implementation yet since there is limited resources available per our research so far). The problem we will be tackling in this project is known as the **Recommendation Systems** problem.

UI example of the desired application:

Movie Recommendation Engine

Choose Three Movies You Like

Toy Story (1995) ▼

Jumanji (1995) ▼

Grumpier Old Men (1995) ▼

Submit

You Might Like These Too!

User-Based Collaborative Filtering Recommended Titles	
1	Smokey and the Bandit II (1980)
2	Eye for an Eye (1996)
3	Princess Caraboo (1994)
4	Drop Dead Fred (1991)
5	Body Parts (1991)
6	Ayn Rand: A Sense of Life (1997)
7	Sense and Sensibility (1995)
8	Heat (1995)
9	Santa with Muscles (1996)
10	Only You (1994)

Source: <https://muffynomster.wordpress.com/2015/06/16/building-an-online-recommender-system/>

Dataset

The dataset comprises ratings of 17,770 movies from 2,649,430 users in the format of 3 columns, i.e., `customer_id`, `customer_rating`, and `movie_id`. Apart from that, we are also provided with a `movie_titles.csv` file which contains the titles of the movies in the dataset, `probe.txt` dataset for validation, and `qualifying.txt` dataset for the actual prediction. This dataset was first introduced by Netflix as a means to improve their recommendation algorithm in a competition which was held from 2006 to 2009. Since the dataset is very large, it will take us a long time to train the model if we opt to include all of the dataset in the training. Hence, we might only focus on some parts of it that we think are essential depending on how long it would take to train the model.

Proposed AI/ML methodologies

There are a lot of different approaches out there to choose from to work on this particular dataset and implement our application. In the following sections, we are going to try to elaborate further regarding their implementations.

1. Collaborative Filtering

This model is a prevalent recommendation technique used for the purpose of making automatic predictions of a user based on his/her similarities in interests, as compared with other users. This model is commonly utilized for giving television show suggestions for users based on its capability to predict which television series the user would like.

2. Pearson's R correlation

Referring to <https://statistics.laerd.com>'s definition on this model, Pearson's R is a measure of a linear correlation strength between two variables. It basically strives to illustrate a best fit line for data of two variables. This model can be used to predict a person's top movie recommendations given a movie that they like.

3. Neural Network

Finally, we might also try to train a standard neural network model as an alternative to a rather conventional Collaborative Filtering approach and see how they compare in terms of prediction accuracy. We will need to experiment with different model

configurations such as hyperparameters and data preprocessing procedures to build the most suitable model for our implementation.

Tools

As of right now, we are still contemplating whether to implement the AI/ML project in R or Python and deploy the application in either of the languages using a UI. Hence, we provide the following libraries that might be suitable for the development of our project in both languages.

1. Python libraries

- Tensorflow (specifically Keras interface)
- Numpy
- Pandas
- Scikit-surprise

2. R libraries

- Tidyverse (for basic data wrangling, and engineering)
- Keras
- Recommenderlab
- Ggplot2
- Caret
- Data.table
- Recosystem
- Shinydashboard

Tasks and Timeline

Timeline	Task	Member
14/03/2021	Data preprocessing	Reynard
21/03/2021	Data exploration and visualization	Reynard, Reno
28/03/2021	Model training and evaluation	Reno
04/04/2021	Error analysis	Reynard
11/04/2021	Model improvement and visualization	Reno

24/04/2021	Project deadline	Reynard, Reno
------------	------------------	---------------

However, in the actual practice, we will be collaborating a lot on each specified task to avoid confusion and maintain coherence within our project implementation.

Referenced materials

Tutorials on developing the application:

Wittenauer, J. (2019, April 29). *Deep Learning With Keras*. johnwittenauer.net.
<https://www.johnwittenauer.net/deep-learning-with-keras-recommender-systems/>

Ivamoto, V. (2019, December 09). *Movie Recommendation System in R*. Rpubs.
<https://rpubs.com/vsi/movielens>

Citations for our proposal:

Chong, D. (2020, April 30). *Deep dive into Netflix's recommender system*.
 towardsdatascience.com.
<https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48>

Laerd Statistics (2020). Pearson's product moment correlation. *Statistical tutorials and software guides*. Retrieved March, 12, 2021, from
<https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php>

Dataset Source:

<https://www.kaggle.com/netflix-inc/netflix-prize-data>