ISSN: 2321-7782 (Online)
e-ISJN: A4372-3114
Impact Factor: 7.327

Volume 6, Issue 5, May 2018

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Movie Recommendation with Machine Learning

Andy W. Chen
Sauder School of Business
University of British Columbia
Vancouver, Canada

Abstract: ***In this paper, I explore the use of a machine learning model, k-nearest neighbors, to find most similar movies. Using a dataset of over 10,000 movies and 1,100 tags of the movies, I rank each movie by similarity to a particular movie in the dataset. The model produces consistent results, finding movies that should be identified as similar by most viewers. The key driver of the model is the accuracy and comprehensiveness of the tags, which cover a wide range of movie characteristics such as places, people, themes, and plot of the movies. The results can be useful for movie recommender systems which are applied by a variety of organizations and businesses such as movies retailers (Amazon, Netflix) and movie database (IMDB).***

Keywords: Movie Industry, Recommendation System, Machine Learning, Data Science, Business Analytics.

I. INTRODUCTION

The movie industry is a growing business, with hundreds of movies released worldwide each year. Many businesses offer recommender systems for movie watchers. For example, Amazon and Netflix provide a list of recommended movies to customers who browse movie products. These recommended movies should be similar to the ones the customer has browsed so the probability of purchase would be high. Websites that provide movie information such as IMDB also provides a list of similar movies a customer would be interested in while browsing.

In this paper, I use a machine learning approach, k-nearest neighbors (KNN) algorithm, to find the most similar movies to each particular movie. The features are tags assigned by movie viewers. Each movie has a relevance score for each tag. The KNN algorithm finds the best number of clusters that will segregate the movies in the data using the relevance scores for each pair movie and tag. I train the KNN model using different number of clusters, and compare each model using the sum of squared errors as the measure. Using this algorithm, I find that the optimal number of clusters is 10. More details about the data and method are described in the paper.

Related work in this area includes the work by Christakou et al.[1], who use semi-supervised learning to build a model for predicting viewer preferences and make recommendations accordingly. Li and Kim[2] use collaborative filtering to build a model to make predictions based on historical data of viewer behavior. Wang et al.[3] used principal component analysis to transform features to be input into the recommender system. Lekakos and Caravelas[4] use a hybrid approach to build a recommender system. Choi et al.[5] use the correlation between genres of movies watched by viewers to enhance their recommendation system.

II. METHOD

I use the MovieLens 20M dataset which consists of 20 million ratings and 465,000 tags applied to 27,000 movies by 138,000 users. The data set also includes a tag genome data with 12 million relevance scores across 1,100 tags. The tags can describe the genre (for example, comedy, drama, thriller), themes (for example, baseball, aviation, college), geography (for example, China,

Africa), sentiments (for example, disappointing, funny), people, objects, and any words or phrases used to describe the movie. My k-nearest neighbors model uses the relevance scores for each movie-tag pair. There are 10,993 movies with relevance scores for 1,128 tags, in total there are 12.4 million relevance scores.

For each movie, The KNN algorithm computes the distance between that particular movie and every other movie. The distance metric used is the Euclidean distance. It can be defined as the square root of the sum of squares of the difference between each feature for a pair of movies. Then for each movie, the KNN model I build gives a list of most similar movies ranked by similarity calculated using the relevance scores.

III. Results and Discussion

The result of the model is a list of most similar movies for each movie. There are over 10,000 movies. I present a sample of the KNN model's output in Table 1. The first row is the movie of interest. Below each movie, there is a list of 20 most similar movies found by the KNN model using the relevance scores. For example, the most similar movie to Toy Story is Monsters, Inc. followed by its sequel, Toy Story 2. The most similar movie to Legally Blonde is Miss Congeniality and the 20th most similar movie to it is My Life in Ruins.

The results look reasonable. For example, the most similar movies to Toy Story are all animated movies, mostly by Disney Pixar. This is because these movies all have similar relevance scores that are high for tags related to topics such as animated movies, comedy, and Disney. For Jumanji, the most similar movies all have the theme of adventure and game element. The KNN model consistently finds the most similar movies.

Table I. 20 Most Similar Movies to Each Movie Using KNN

Toy Story (1995)	**Jumanji (1995)**	**Devil Wears Prada, The (2006)**	**Kong: Skull Island (2017)**	**Legally Blonde (2001)**
Monsters, Inc. (2001)	Honey, I Shrunk the Kids (1989)	Nanny Diaries, The (2007)	Godzilla (2014)	Miss Congeniality (2000)
Toy Story 2 (1999)	Mighty Joe Young (1998)	Confessions of a Shopaholic (2009)	Face/Off (1997)	What Women Want (2000)
Bug's Life, A (1998)	Zathura (2005)	Sex and the City (2008)	Predator (1987)	How to Lose a Guy in 10 Days (2003)
Ratatouille (2007)	Night at the Museum (2006)	Morning Glory (2010)	King Kong (2005)	13 Going on 30 (2004)
Toy Story 3 (2010)	Borrowers, The (1997)	First Wives Club, The (1996)	The Magnificent Seven (2016)	House Bunny, The (2008)
Finding Nemo (2003)	Small Soldiers (1998)	I Don't Know How She Does It (2011)	Expendables, The (2010)	Never Been Kissed (1999)
Ice Age (2002)	Free Willy (1993)	Bridget Jones's Baby (2016)	Predators (2010)	Head Over Heels (2001)
Luxo Jr. (1986)	Gnome-Mobile, The (1967)	Waiting to Exhale (1995)	Rambo (Rambo 4) (2008)	Bring It On (2000)
Shrek (2001)	Escape to Witch Mountain (1975)	How to Be Single (2016)	Pacific Rim (2013)	She's All That (1999)
The Lego Movie (2014)	Water Horse: Legend of the Deep, The (2007)	Eat Pray Love (2010)	Con Air (1997)	Princess Diaries, The (2001)
How to Train Your Dragon (2010)	Twister (1990)	He's Just Not That Into You (2009)	Jurassic World (2015)	Win a Date with Tad Hamilton! (2004)
Wreck-It Ralph (2012)	Santa Clause, The (1994)	Hope Springs (2003)	Hercules (2014)	View from the Top (2003)
Lion King, The (1994)	Computer Wore Tennis Shoes, The (1969)	Banger Sisters, The (2002)	First Blood (Rambo: First Blood) (1982)	Shallow Hal (2001)
Incredibles, The (2004)	Nim's Island (2008)	In Her Shoes (2005)	Universal Soldier: Regeneration (2009)	Guy Thing, A (2003)
Up (2009)	Kid, The (2000)	Raising Helen (2004)	Spectral (2016)	Failure to Launch (2006)
Aladdin (1992)	Santa Claus: The Movie (1985)	Someone Like You (2001)	Cloverfield (2008)	John Tucker Must Die (2006)
Cars (2006)	Flintstones, The (1994)	Something Borrowed (2011)	Live Free or Die Hard (2007)	Sydney White (2007)
Monsters University	Flubber (1997)	Best Man, The (1999)	John Wick: Chapter	Freaky Friday (2003)

(2013)			Two (2017)	
Antz (1998)	Mighty Joe Young (1949)	Uptown Girls (2003)	Rise of the Planet of the Apes (2011)	Love Potion #9 (1992)
Iron Giant, The (1999)	Tall Tale (1995)	America's Sweethearts (2001)	Poseidon Adventure, The (1972)	My Life in Ruins (2009)

IV. Conclusion

This paper shows the use of k-nearest neighbors, a machine learning model, to find the most similar movies. The results are consistent across the movies. A big factor for the accuracy is the comprehensiveness of the tags, which cover a variety of topics. The results of the model can be useful for movie recommender systems to predict viewer preferences and make suitable recommendations for better customer experience. A potential future extension could be to incorporate viewer history to improve the prediction.

References

1. Christakou C, Lefakis L, Vrettos S, Stafylopatis A. A Movie Recommender System Based on Semi-Supervised Clustering. International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). 2005;897-903.
2. Li Q, Kim BM. Clustering Approach for Hybrid Recommender System. Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003). 2003; 33-38.
3. Wang Z, Yu X, Feng N, Wang Z. An Improved Collaborative Movie Recommendation System Using Computational Intelligence. Journal of Visual Languages & Computing. 2014;25(6):667-675.
4. Lekakos G, Caravelas P. A Hybrid Approach for Movie Recommendation. Multimedia Tools and Applications. 2008;36(1-2):55-70.
5. Choi SM, Ko SK, Han YS. A Movie Recommendation Algorithm Based on Genre Correlations. Expert Systems with Applications. 2012;39(9):8079-8085.

