Netflix Recommendation System with Python

[1]Pannala Meghana,[2]Rajnish Kumar,[3]CH. Narasimha Chary
[1,2]UG Scholars, [3]Assistant Professor
[1,2,3]Department of Computer Science and Engineering
[1,2,3]Guru Nanak Institutions Technical Campus, Hyderabad, Telangana, India.

***ABSTRACT*:** The use of a Recommendation system is to provide users with recommendations based on their search preferences. In this article, I will introduce you to a machine learning project on the Netflix recommendation system with Python. Netflix is a company that manages a large collection of TV shows and movies, streaming it anytime via online. This business is profitable because users make a monthly payment to access the platform. However, customers can cancel their subscriptions at any time. Therefore, the company must keep the users hooked on the platform and not lose their interest. This is where recommendation systems start to play an important role, providing valuable suggestions to users is essential. Netflix's recommendation system helps them increase their popularity among service providers as they help increase number of items sold, offer a diverse selection of items, increase user satisfaction, as well as user loyalty to the company, and they are very helpful in getting a better understanding of what the user wants. Then it's easier to get the user to make better decisions from a wide variety of movie products. The idea behind the Netflix recommendation system is to recommend the most popular movies to users. They could be the most-watched, or also the ones with the highest ratings. The popularity of recommendations can be built based on usage data and article content.

Keywords: Pearson R Correlation, Collaborative Filtering, Content-Based Filtering, Data Analysis.

I. INTRODUCTION

Netflix is a company that handles a big collection of television programs and movies, by streaming it at any time via online (computers or TV). This firm is profitable because the users do a monthly payment to get access to the platform. However, the clients can cancel their subscriptions at any time (Amatriain, 2013). Therefore, it is vital for the business to keep the users hooked to the platform and not to lose their interest. This is where recommendation systems start to play an important role, it is pivotal to provide valuable suggestions to users (Ricci et al., 2010).

The recommendation systems are increasing their popularity among the service providers, because they help to increase the number of items sold, offer a diverse selection of items, the user satisfaction increases, as well as the user fidelity to the company, and they are quite helpful to have a better understanding of what the user wants (Ricci et al., 2010). Then, it is easier to lead the user to make better decisions from a wide variety of cinematographic products.

The recommender systems take into account not only information about the users but also about the items they consume; comparison with other products, and so on and so forth (Hahsler, 2014). Nevertheless, there are many algorithms available to perform a recommendation system. For instance, (i) Popularity, where only the most popular items are recommended (ii) Collaborative Filtering, which looks for patterns in the user activity to produce user-specific recommendations (Breese, Heckerman, and Kadie, 1998); (iii) Content-based Filtering, the recommendation of items with similar information the user has liked or used in the past (description, topic, among others) (Aggarwal, 2016) ; (iv) Hybrid Approaches, combines the two algorithms mentioned above (Adomavicius and Tuzhilin, 2005).

II. EXISTING SYSTEM :

The existing system for Netflix movie recommendations implements a traditional recommendation framework to predict and suggest top movies to users. This framework relies on algorithms such as content-based filtering and collaborative filtering, which analyze user preferences, viewing history, and movie attributes. To enhance prediction accuracy, Netflix employs a hybrid approach, combining the strengths of both content-based and collaborative filtering. This ensures a comprehensive understanding of user preferences and results in highly personalized recommendations.

DISADVANTAGES:

- Less accuracy
- Difficult to predict

III. PROPOSED SYSTEM :

In the proposed system, we aim to implement a personalized recommendation system for Netflix using advanced machine learning techniques. The core of the recommendation system is based on the Pearson R Correlation, a statistical measure used to determine the strength and direction of the relationship between two variables.

ADVANTAGES:

- Better accuracy
- Easy to predict

IV. LITERATURE SURVEY:

A) **TITLE:** Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.

AUTHOR: Adomavicius, Gediminas and Alexander Tuzhilin.

YEAR: 2005.

Summary:

This paper presents an overview of the field of recommender systems and describes the current generation of recommendation methods that are usually classified into the following three main categories: content-based, collaborative, and hybrid recommendation approaches. This paper also describes various limitations of current recommendation methods and discusses possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. These extensions include, among others, an improvement of understanding of users and items, incorporation of the contextual information into the recommendation process, support for multi criteria ratings, and a provision of more flexible and less intrusive types of recommendations.

B) TITLE: : Mining Large Streams of User Data for Personalized Recommendations.
AUTHOR: Amatriain, Xavier.
YEAR: 2013.

Summary:
The Netflix Prize put the spotlight on the use of data mining and machine learning methods for predicting user preferences. Many lessons came out of the competition. But since then, Recommender Systems have evolved. This evolution has been driven by the greater availability of different kinds of user data in industry and the interest that the area has drawn among the research community. The goal of this paper is to give an up-to-date overview of the use of data mining approaches for personalization and recommendation. Using Netflix personalization as a motivating use case, I will describe the use of different kinds of data and machine learning techniques. After introducing the traditional approaches to recommendation, I highlight some of the main lessons learned from the Netflix Prize. I then describe the use of recommendation and personalization techniques at Netflix. Finally, I pinpoint the most promising current research avenues and unsolved problems that deserve attention in this domain.

C) TITLE: Data Mining Methods for Recommender Systems

AUTHOR: Aggarwal, Charu C.

YEAR: 2011.

Summary:
In this chapter, we give an overview of the main Data Mining techniques used in the context of Recommender Systems. We first describe common preprocessing methods such as sampling or dimensionality reduction. Next, we review the most important classification techniques, including Bayesian Networks and Support Vector Machines. We describe the k-means clustering algorithm and discuss several alternatives. We also present association rules and related algorithms for an efficient training process. In addition to introducing these techniques, we survey their uses in Recommender Systems and present cases where they have been successfully applied.

V. METHODOLOGIES

MODULES NAME:

1. Data Gathering
2. Data Preparation
3. Data Reduction
4. Recommendation Algorithm
5. Prediction

MODULES EXPLANATION

- **Data Gathering**

To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be comprised of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases. For instance, a business dataset will be entirely different from a medical dataset. In our project the dataset gives the Kaggle and it has a dataset namelymovie_titles.csv.

- **Data Preparation**

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data and the combining of data sets to enrich data. The Movie ID column is a mess. We're going to improve this by first creating a NumPy array with the correct length, then adding the entire array as a column in the main data frame.

- **Data Reduction**

Data reduction is the process of reducing the amount of capacity required to store data. Data reduction can increase storage efficiency and reduce costs. Storage vendors will often describe storage capacity in terms of raw capacity and effective capacity, which refers to data after the reduction. The dataset is super huge. Therefore, we need to reduce the data volume by improving data quality. We can use these two approaches:

  - ✓ Delete the movie with too many reviews (they are relatively unpopular).
  - ✓ Remove the customer who gives too less notice (he is relatively less active).

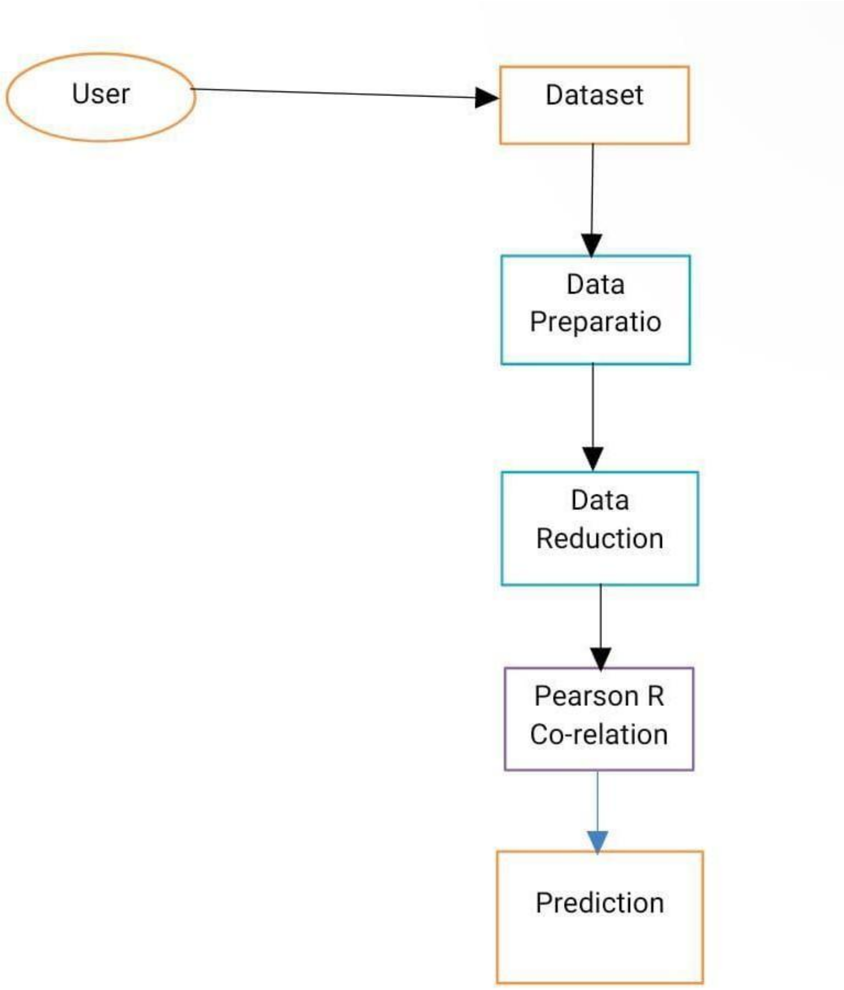
- **Recommendation Algorithm**

The Pearson correlation coefficient (named for Karl Pearson) can be used to summarize the strength of the linear relationship between two data samples. The Pearson's correlation coefficient is calculated as the covariance of the two variables divided by the product of the standard deviation of each data sample. Now let's build the Netflix recommendation system by using Pearson correlation algorithm.

- **Prediction**

Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome. In this project, it will predict the top10 movies based on Pearson correlation algorithm.

VI. SYSTEM ARCHITECTURE



System Architecture Model

VII. CONCLUSION

In this work, we have proposed a new effective method for synthetic aging of human faces based on Age Conditional Generative Adversarial Network (Age-cGAN). The method is composed of two steps: (1) input face reconstruction requiring the solution of an optimization problem in order to find an optimal latent approximation $z*$, (2) and face aging it-self performed by a simple change of condition y at the input of the generator. The cornerstone of our method is the novel "Identity-Preserving" latent vector optimization approach al-lowing to preserve the original person's identity in the reconstruction. This approach is universal meaning that it can be used to preserve identity not only for face aging but also for other face alterations (e.g. adding a beard, sunglasses etc.

Our face aging method can be used for synthetic augmentation of face datasets and for improving the robustness of face recognition solutions in cross-age scenarios.

VIII. REFERENCES

1. Adomavicius, Gediminas and Alexander Tuzhilin (2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". In: IEEE Trans. on Knowl. and Data Eng. 17.6, pp. 734–749. ISSN: 1041- 4347. DOI: 10.1109/TKDE.2005.99. URL: https://doi.org/10.1109/TKDE.2005. 99.
2. Aggarwal, Charu C. (2016). Recommender Systems: The Textbook. 1st. Springer Publishing Company, Incorporated. ISBN: 3319296574, 9783319296579. Amatriain, Xavier (2013). "Mining Large Streams of User Data for Personalized Recommendations". In: SIGKDD Explor. Newsl. 14.2, pp. 37–48. ISSN: 1931-0145. DOI: 10.1145/2481244.2481250. URL: http://doi.acm.org/10.1145/2481244. 2481250.
3. Amatriain, Xavier et al. (2011). "Data Mining Methods for Recommender Systems". In: Recommender Systems Handbook. Ed. by Francesco Ricci et al. Boston, MA: Springer US, pp. 39–71. ISBN: 978-0-387-85820-3. DOI: 10 . 1007 / 978 - 0 - 387 - 85820-3_2. URL: https://doi.org/10.1007/978-0-387-85820-3_2.
4. Baeza-Yates, Ricardo, Berthier Ribeiro-Neto, et al. (1999). Modern information retrieval. Vol. 463. ACM press New York. Bokde, Dheeraj, Sheetal Girase, and Debajyoti Mukhopadhyay (2015). "Matrix factorization model in collaborative filtering algorithms: A survey". In: Procedia Computer Science 49, pp. 136–146.
5. Breese, John S., David Heckerman, and Carl Kadie (1998). "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. UAI'98. Madison, Wisconsin: Morgan Kaufmann Publishers Inc., pp. 43–52. ISBN: 1-55860-555-X. URL: http://dl.acm.org/citation.cfm?id=2074094.2074100.
6. Bressan, M. et al. (2016). "The Limits of Popularity-Based Recommendations, and the Role of Social Ties". In: ArXiv e-prints. arXiv: 1607.04263.
7. Girase, Sheetal, Debajyoti Mukhopadhyay, et al. (2015). "Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey". In: arXiv preprint arXiv:1503.07475.
8. Gunawardana, Asela and Guy Shani (2009). "A survey of accuracy evaluation metrics of recommendation tasks". In: Journal of Machine Learning Research 10.Dec, pp. 2935–2962.
9. Hahsler, Michael (2014). recommender lab: Lab for Developing and Testing Recommender Algorithms. R package version 0.1-5. URL: http://CRAN.R-project.org/package= recommender lab.
10. Isinkaye, F.O., Y.O. Folajimi, and B.A. Ojokoh (2015). "Recommendation systems: Principles, methods and evaluation". In: Egyptian Informatics Journal 16.3, pp. 261 –273. ISSN: 1110-8665. DOI: https://doi.org/10.1016/j.eij.2015.06.005.

11. URL: http://www.sciencedirect.com/science/article/pii/S1110866515000341. Kalman, Dan (1996). "A singularly valuable decomposition: the SVD of a matrix". In: The college mathematics journal 27.1, pp. 2–23.applications," Tech. Rep., CMU-CS-16-118, CMU School of Computer Science, 2016.
12. Liang, Xijun et al. (2016). "Measure prediction capability of data for collaborative filtering". English. In: Knowledge and Information Systems 49.3. Copyright - Springer- Verlag London 2016; Last updated - 2016-11-03; CODEN - KISNCR, pp. 975– 1004. URL: https://search- proquest- com.vu-nl.idm.oclc.org/docview/ 1828122760?accountid=10978.
13. Melville, Prem, Raymond J. Mooney, and Ramadass Nagarajan (2002). "Content- Boosted Collaborative Filtering for Improved Recommendations". In: Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02). Edmonton, Alberta, pp. 187–192. URL: http://www.cs.utexas.edu/users/ailab/? melville:aaai02.
14. Mobasher, Bamshad et al. (2005). "Effective attack models for shilling item-based collaborative filtering systems". In: Proceedings of the 2005 WebKDD Workshop, held in conjunction with ACM SIGKDD. Vol. 2005.
15. Ricci, Francesco, Lior Rokach, and Bracha Shapira (2011). "Introduction to recommender systems handbook". In: Recommender systems handbook. Springer, pp. 1– 35.
16. Ricci, Francesco et al. (2010). Recommender Systems Handbook. 1st. New York, NY, USA: Springer-Verlag New York, Inc. ISBN: 0387858199, 9780387858197.
17. Salton, Gerard (1989). "Automatic text processing: The transformation, analysis, and retrieval of". In: Reading: Addison-Wesley.
18. Sarwar, Badrul et al. (2001). "Item-based Collaborative Filtering Recommendation Algorithms". In: Proceedings of the 10th International Conference on World Wide Web. WWW'01. Hong Kong, Hong Kong: ACM, pp. 285–295. ISBN: 1-58113-348-0. DOI: 10.1145/371920.372071. URL: http://doi.acm.org/10.1145/371920.372071.
19. Schafer, J. Ben, Joseph Konstan, and John Riedl (1999). "Recommender Systems in e-Commerce". In: Proceedings of the 1st ACM Conference on Electronic Commerce. EC '99. Denver, Colorado, USA: ACM, pp. 158–166. ISBN: 1-58113-176-3. DOI: 10.1145/336992.337035. URL: http://doi.acm.org/10.1145/336992.337035.
20. Zhang, Tong and Vijay S Iyengar (2002). "Recommender systems using linear classifiers". In: Journal of Machine Learning Research 2.Feb, pp. 313–334.