IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

RECOMMENDATION SYSTEM USING KNN AND COSINE SIMILARITY

PROF. SHIVGANGA GAVHANE
OF (SPPU) COMPUTER ENGINEERING,
DYPIEMR, AKURDI.

JAYESH PATIL
OF (SPPU) COMPUTER ENGINEERING,
DYPIEMR,AKURDI

HARSHAL KADWE
OF (SPPU) COMPUTER ENGINEERING,
DYPIEMR,AKURDI.

PRAJWAL THACKREY
OF (SPPU) COMPUTER ENGINEERING
DYPIEMR,AKURDI

SUSHOVAN MANNA
OF (SPPU) COMPUTER ENGINEERING,
DYPIEMR, AKURDI.

Abstract: After looking at the needs of today's generation and high demand for similar product utilization people find difficulty in sorting for product in e-commerce website but product recommendation does it in very accurately and recommends what exactly the user wants. Machine learning is used now-a-days in many area to improve recommendation techniques rather than using filtering techniques. In advance technology, people are more open minded and rely heavily on modern applications for buying accessories, for watching movies and stuffs related to their daily needs. Due to this perpetual rise in online shopping demand and movie browsing , the organizations are relying on machine learning based technologies which are helping them in seeking the actual targeted users with less efforts as compared to earlier methods of advertisement. We are developing this Technology which helps us to understand the requirements and gives recommendation for the product searched by the user. This model compares various machine learning algorithms for recommendation of various product buying pattern by users and gives more accurate result related to search.

Keywords: Machine learning, recommendation systems, Supervised, Unsupervised Learning, KNN Algorithm , Cosine Similarity ,Collaborative Filtering.

I. INTRODUCTION

In this world ,the Web is the most essential part and many technologies are developed on this to earn and get rewards contributing in their own domain. In the same manner people's activities-Internet users open more opportunity to gain more information about peoples likes and dislikes. This certain amount of information is used as knowledge, understanding the pattern of the users it gives the idea to recommend the similar products they need. This allows you to monitor network user's various activities without affecting their intention. These are valuable insights for the marketing market, for example the marketplace. They show whether their sales tactics are working for a given social group, and sometimes they help to find the best way to reach a given target group. Working with knowledge base assessment it is difficult for scientist and application developers to work on data mining. The machine learning problem comes with support. Statistical methods can be one of the fundamental techniques in machine learning: regression and study of association. More advanced methods are problems associated with learning neural networks or fuzzy logic. The designer creates a recommendation algorithm and the computer determines the conclusion related to the properties of this set on its basis, acting on a given set of data. There are great opportunities for maximize profit.

II. MACHINE LEARNING

ALGORITHM USED IN MODEL

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

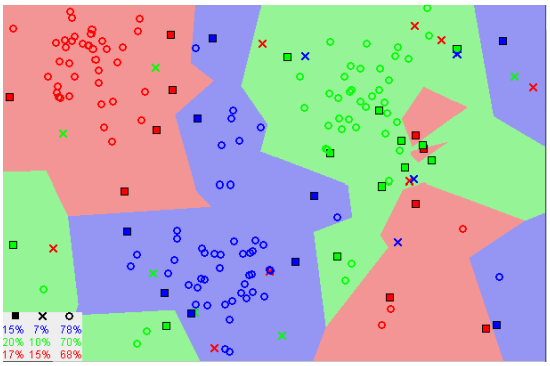


Fig1.Image showing how similar data points typically exist close to each other

Remember in the above picture that identical data points are closely linked much of the time. The KNN algorithm relies on this assumption as being valid enough to be useful to the algorithm. KNN captures the concept of similarity (sometimes referred to as distance, proximity or closeness) with some of the mathematics calculating the distance between points on a graph.
There are other ways to measure distance, and one way might be better depending on what problem we solve. The straight-line distance (also known as the Euclidean distance) is however a common and familiar choice.

The KNN Algorithm

1.Load the data
2.Initialize K to your chosen number of neighbors
3. For each example in the data
    3.1 Calculate the distance between the query example and the current example from the data.
    3.2 Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the mode of the K labels

Choosing the right value for K :-
To choose the K that is right for your data, we run the KNN algorithm multiple times with different K values and choose the K that reduces the number of errors we encounter while retaining the capacity of the algorithm to make predictions correctly when given data that it has not seen before.
Below are only a few things to remember:

1.When we lower the K-value to 1, our predictions are less accurate. Just think for a minute, imagine K=1 and we've got a question point surrounded by several reds and one green (I'm talking about the colored plot's top left corner above), but the green is the nearest neighbor. Reasonably we'd think the
question point is most likely red, but since K=1, KNN predicts incorrectly predicts a Green Query Point.

2. Conversely, as we increase K's value, our predictions become more reliable due to majority vote / average, and thus more likely to make predictions more accurate (up to a certain point). Eventually we start seeing an growing abundance of errors. It's here that we think we've taken K's worth too far.

3. In cases where we take a majority vote among labels (e.g., picking the mode in a classification problem), we typically make K an odd number to have a tiebreaker.

KNN in practice

In practice KNN's key downside of being slightly slower as data volume increases makes it an inefficient alternative in environments where predictions need to be made quickly. In addition, faster algorithms can generate more accurate results in the classification and regression. Nonetheless, as long as you have enough computational power to handle the data that you use to make predictions quickly, KNN can also be useful in solving problems that have solutions that rely on identifying similar artifacts. An example of this is using the KNN algorithm, an implementation of KNN-search in recommender system.

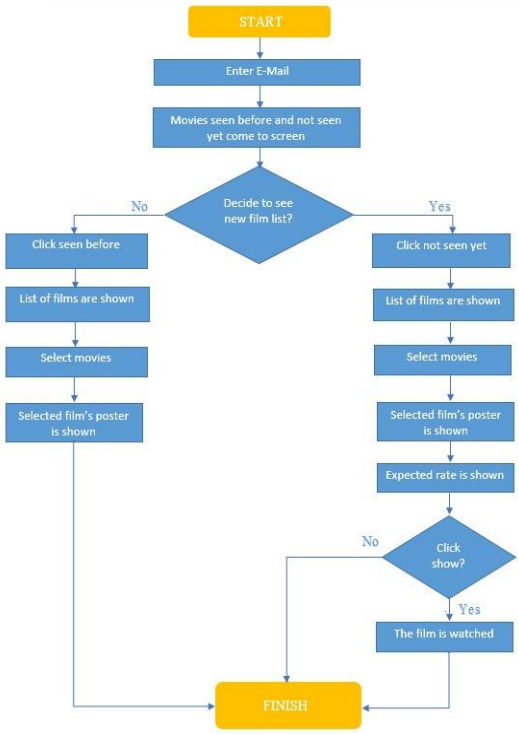
COSINE SIMILARITY:

Cosine similitude tests the similarity of an inner product space between two vectors. It is determined by the angle cosine between two vectors, which defines when two vectors point in approximately the same direction. It is also used in text analysis to measure document similitude. A document may represent thousands of attributes, each documenting the frequency of a given w.

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

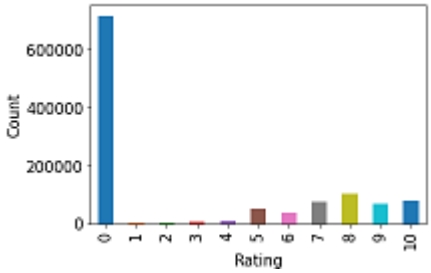
IV.EXISTING SYSTEM

The existing system presented does not utilizes the collaborative and KNN as one unit. Hence the performance of the proposed system is comparatively higher and consistent. The existing system utilizes content based filtering which is not that reliable and consistent with its performance in longer run with larger datasets. Due to content based filtering the model needs to retrain itself even for an inconsiderable amount of data. Because of which the computational expenses increases drastically and becomes way to costlier for the application and its utilization in small scale businesses.

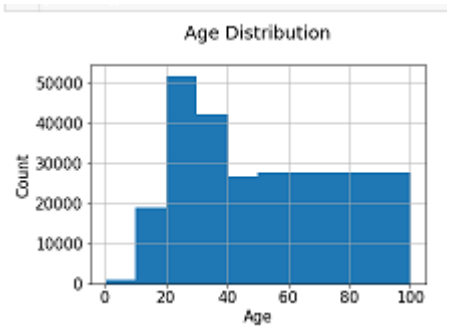


V. PROPOSED MODEL

For coming up with more unique solution we have utilized an approach which utilizes both KNN algorithm and cosine similarity for collaborative filtering. The huge amount of data is gathered from the IIF(Institute for information Freiburg) related with books. The dataset varies around 0.6 million in quantity. Hence, it was challenging to utilize the most relevant data entries. For that we replaced the entries having ratings less than 100 people and also the user ids who gave less than 200 ratings to the dataset. The ratings to it were given between 0 to 10 by various age categories out of which people ranging from 20 to 40 gave the maximum ratings. Even though we haven't given any parameter to reduce dataset on the basis of age. It is just for information purpose only. The images related to the analysis are given below



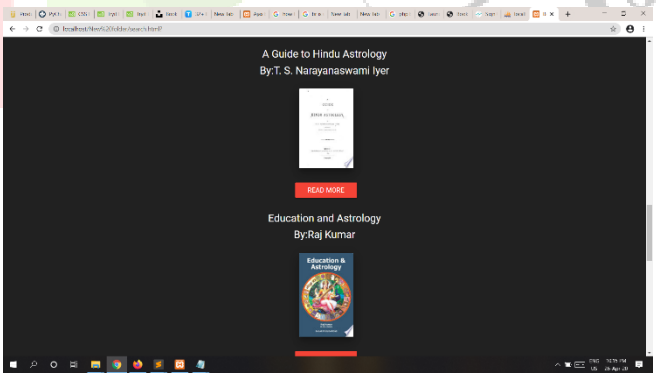
Here it can be clearly noticed that the books are rated between 0 to 10. Even the ratings given by different age group is given below fig.



After having all the analysis the average rating count is taken out and again the threshold value of 50 is taken to remove more irrelevant data from the set. The next procedure is followed by creating matrix of user ratings for all the books vs book title names. Then the individual cell acts as a vector in dataset and it is here we utilize cosine similarity to find the similarity between the by having their dot product. But even this is not the end the vector points are also considered as fixed points on data-plane, the KNN comes handy in here and also finds the Euclidean distance between these data points as per the value of K. As it is evident in many cases after studies that collaborative based recommendation has much better performance as compared to the content based filtering. Even when the huge amount of dataset is utilized it comes more handy and requires less computation resources. The model isn't trained again until considerable amount of data is gathered again from the users. Until then a binary file of trained model is formed in a joblib file which is used as ready to use recommending model. The data utilized is in multiple csv (comma separated values) files. This files are then merged with a common attribute in them viz. ISBN which is a unique id given to each book.

VI. RESULTS

The book recommender system created using KNN and cosine similarity provides more relevant recommendations to the users. The web application is made for the book store using javascript , html, css, etc. The dataset utilized is also refined in multiple procedures to provide more accurate results viz. threshold value elimination.



The recommend button is present on the navigation bar which provides recommendations of the input search given by the user.

VII.FUTURE SCOPE

Even though the computational expenses is less as compared to the deep learning methods and neural network technology. It serves a very useful application in the region where the systems work on limited resources and expenditure. Leading to more demands in small scale businesses, grocery shops, stores like crossword. The approach can be further improved by using more resource expensive ways viz. DNN, Tensor flow ,etc.

VIII.ACKNOWLWDGEMENTS

The project was only possible due to constant support and guidance of Prf. Shivganga Gavhane and team members. Also the encouragement provided by our H.O.D. ma'am Prof. Pratiksha Shevatekar. Due to time to time support and evaluation of our mistakes this paper has been possible and lead to working with our potential as a team collaboration integrally.

REFERENCES

1. Mohammed FadhelAljunid and Manjaiah D.H."A SURVEY ON RECOMMENDATION SYSTEMS FOR SOCIAL MEDIA USING BIG DATA ANALYTICS". International Journal of Latest Trends in Engineering and Technology, Special Issue (SACAIM 2017), pp.48-58.
2. S. Taneja, C. Gupta, D. Gureja and K. Goyal, "K Nearest-Neighbor Techniques for Data Classification-AReview," Proc. International Conference on Computing, Informatic and Network (ICCIN2K14), Jan. 2014, pp. 69-73.
3. H. Hong, G. Juan and W. Ben, "An Improved KNN Algorithm Based on Adaptive Cluster Distance Bounding for High Dimensional Indexing," Proc. 3rd Global Congress on Intelligent Systems (GCIS), Conference Publishing Services, Nov. 2012, pp. 213-217, doi:10.1109/GCIS.2012.86.
4. J. Gou, L. Du, Y. Zhang and T. Xiong, "A New Distance-weighted k-nearest Neighbor Classifier," Proc. Journal of Information & Computational Science, June 2012, pp. 1429-1436.
5. C. De Boom, S. Van Canneyt, S. Bohez, T. Demeester, and B. Dhoedt, "Learning Semantic Similarity for Very Short Texts," 2015.
6. A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," Expert Syst. Appl., vol. 41, no. 4, pp. 1432–1462, Mar. 2014.
7. G. Adomavicius, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", IEEE Transactions on Knowledge and Data Engineering, vol 17, no. 6, 2005.
8. W. Yang, Z. Wang, M. You, "An improved collaborative filtering method for recommendations' generation", in Proc. of IEEE International Conference on Systems, Man and Cybernetics, 2004.