RECOMMENDATION SYSTEM USING CLUSTERING AND NEURAL NETWORKS

A project report submitted in partial
fulfillment of the requirements for the degree of
Bachelor of Technology
in

Computer Science & Engineering

by

Pallavi Rana and Sonam Gupta

Under the supervision of

Prof. Dr. Satyajit Chakraborty
Director
Institute of Engineering and Management

and

Prof. Dr. Saptarsi Goswami
Assistant Professor
University of Calcutta

IEM
श्रद्धावान लभते ज्ञानम्
INSTITUTE OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

Department of Computer Science & Engineering
Institute of Engineering & Management
Gurukul, Y-12, EP Block, Sector-V, Salt Lake Electronics Complex
Kolkata-700091, West Bengal, India

May, 2018

ABSTRACT

Recommender systems are widely used nowadays, especially in electronic commerce and social networks. We have a huge amount of information overloaded over Internet. It becomes a herculean task for the user to get the relevant information. To some extent, the problem is being solved by the search engines, but they do not provide the personalization of data. So, for filtering this information, we need a recommendation engine. Recommender system use artificial intelligence methods to provide users with recommendation.This paper proposed a machine learning approach to recommend movies to users using K-means clustering algorithm to separate similar users and creating a neural network for each cluster.

ACKNOWLEDGEMENTS

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guides and mentor **Prof. Dr. Saptarsi Goswami** of University of Calcutta for his wisdom, guidance and inspiration that helped us go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to **Prof. Satyajit Chakraborty**, Director, **Prof. Dr. Amlan Kusum Nayak**, Principal and **Prof. Dr. Debika Bhattacharyya**, HOD of Computer Science & Engineering and other faculties of Institute of Engineering & Management, for their assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Pallavi Rana
Reg No. 141040110097 of 2014-2015
Sonam Gupta
Reg No. 141040110173 of 2014-2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1 Motivation

Recommendation systems help users find items (e.g., books, movies, restaurants) from the huge number available on the internet. Given a large set of items and a description of the user's needs, they present to the user a small set of the items that are well suited to the description. Differences in personal preferences, social and educational backgrounds, and private or professional interests are pervasive. As a result, it seems desirable to have *personalized* intelligent systems that process, filter, and display available information in a manner that suits each individual using them. The need for personalization has led to the development of systems that adapt themselves by changing their behavior based on the inferred characteristics of the user interacting with them.

In this paper, we have proposed a machine learning based way to recommend movies using clustering and machine learning approaches.

1.2 Objective

The objective of this project is to distinguish users using clustering algorithm in order to find users with similar taste of movies. Machine learning approaches are used to guess what rating a particular user might give to a particular movie so that this information can be used to recommend movies to viewers.

In this recommendation system, we used the publicly available MoveLens [1] dataset.
The MovieLens dataset contains information about movie ratings by users, information about movies and users. The number of movies in the dataset is 1682 and number of users is 943. The movie data has been split based on their genre and later outer joined with ratings of movies in order to get user preference, average rating and consumption ratio for each genre of movies in three separate approaches. This resulted in each tuple having 18 attributes in all the approaches since there are 18 genres listed in MovieLens[1] dataset. Clustering was used to separate dissimilar users and the result was compared in the three approaches to choose the best one. Principal Component Analysis [2] (PCA) was used to decrease the dimension for

a better clustering result. Feature scaling was done to normalize the input matrix. Then for each cluster, the ratings of the movies were predicted using a neural network. Recommendations were made based on the average rating given to different movies by a user and the predicted rating for a user.

1.3 Organization

CHAPTER 1. INTRODUCTION: Gives a brief introduction on what is being done and why this topic has been chosen.

CHAPTER 2. BACKGROUND: In this section we shall discuss a literature survey of projects done on similar topics and how the authors tried to achieve their objective.

CHAPTER 3. PROPOSED FRAMEWORK: In this section, the methods of how to go about the entire technique is described.

CHAPTER 4. EXPERIMENTAL RESULTS AND ANALYSIS: This section produces a report of how our framework is performing.

CHAPTER 5. CONCLUSION: The limitations of our framework and future scope of the work i.e. where and how we can improve this technique to make it more suitable.

CHAPTER 2. BACKGROUND

2.1 Basic Recommendation System Concepts

Recommender uses three filtering method to filter a large set of data .Using dataset it analyze data and gives fast prediction. Three filtering methods are as follows:

A. Collaborative filtering[3]
Collaborative filtering methods are based on collecting and analyzing a large amount of information on users’ behaviours, activities or preferences and predicting what users will like based on their similarity to other users. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself.

B. Content-based filtering[3]
These algorithms try to recommend items that are similar to those that a user liked in the past. It gathers all the information about the user content and recommends similar attributes content.

C. Hybrid models[3]
This model combines content based prediction as well as collaborative filtering prediction to give a separate result. Netflix is a good example of the use of hybrid recommender Systems

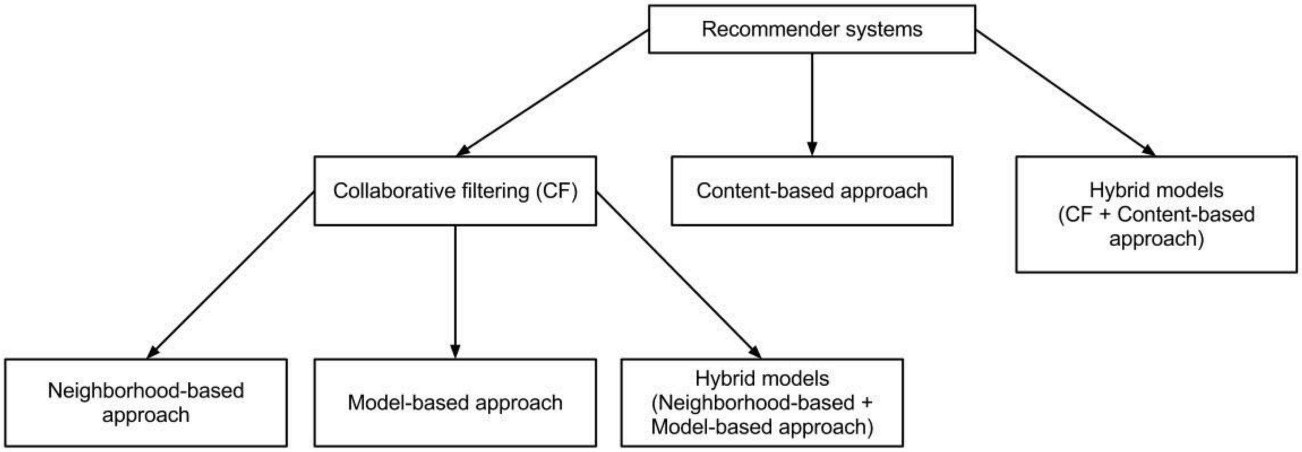

Recommender systems
Collaborative filtering (CF)
Content-based approach
Hybrid models (CF + Content-based approach)
Neighborhood-based approach
Model-based approach
Hybrid models (Neighborhood-based + Model-based approach)

Fig.1: Classification of Recommendation Systems

2.2 Related Literature Review

The reading process was focused on finding three pieces of information: the machine learning algorithm, the domain of the case study or implementation, and the problems or open questions associated with the proposed work. Therefore, after reading the abstract and the introduction, the authors read the approach and case study description, and finally the conclusion or future work of each publication included in the systematic review.

Table 1. Various fields in which recommendation system algorithm is used:

Domain	Publication	Total
Movie	[22][22][22][8][11][23][25][25][16][16]	10
Documents	[28][12][13][13]	4
Product review	[20][29][18][21]	4
Jokes	[8][16]	2
Music	[8][25]	2
TV	[21][30]	2

Academic	[26]	1
Books	[11]	1
Elections	[31]	1
E-mail	[19]	1
E-shop	[9]	1
Mobile navigation	[10]	1
Safety-critical process automation	[17]	1
Social	[24]	1
Stocks	[32]	1
Telecommunication	[17]	1
Tourism	[14]	1
Webpages	[15]	1

Table 2. Machine learning algorithms used in recommendation systems:

Categories	**Publications**	**Total**
Decision Tree	[9][11][23][27][18]	5
Matrix Factorization based	[33][25][16][26]	4
Neighbor-based	[20][24][23][16]	4
Neural Network	[22][9][23][7]	4
Rule Learning	[19][20][11][21]	4
Ensemble	[17][9][18]	3

Gradient descent-based	[15][8][16]	3
Kernel methods	[8][9][14]	3
Clustering	[12][13]	2
Associative classification	[11]	1
Bandit	[10]	1
Lazy learning	[9]	1
Regularization methods	[7]	1

C. Literature survey

1. Neighbourhood based approach[2]:

It also referred as memory- based algorithms. It is based on the fact that similar users display similar patterns of rating behavior and similar items receive similar ratings.

- *User-based collaborative filtering:*

In this filtering, the ratings provided by similar users to a target user A are used to make recommendations for A. The predicted ratings of A are computed as the weighted average values of these “peer group” ratings for each item.

- *Item –based collaborative filtering [6]:*

In order to make recommendations for target itemB, the first step is to determine a set S of items, which are most similar to item B. Then, in order to predict the rating of any particular user A for item B, the ratings in set S, which are specified by A, are determined. The weighted average of these ratings is used to compute the predicted rating of user A for item B.
Thus neighborhood based approach have advantage that they can be combined with other optimization models for better prediction .This approach also face numerous challenges because of data sparsity. This approach is not fast and scalable, thus Model-based approach is preferred over this.

2. Model-based approach [1]:

This approach extracts some information from the dataset, and uses that as a "model" to make recommendations without having to use the complete dataset every time. This approach potentially offers the benefits of speed , scalability and avoidance of overfitting .There are model based approach collaborative filtering algorithm such as Bayesian network ,clustering models ,latent semantic models such as singular value decomposition ,probabilistic latent semantic analysis ,multiple multiplicative factor, latent dirichlet allocation and Markov decision process based models. It suffers from inflexibility and quality of prediction.

- *Hybrid models:*

Four major recommendation techniques constructing hybrids are collaborative filtering, content-based , demographic, and knowledge-based.

Here are some examples of recent hybrid models:

1. A graph based recommender approach which combines content-based and collaborative approach (Huang, Z., et. al, 2002)
2. Item Based Clustering Hybrid Method (ICHM) (Li, Q., and Kim, B.M., 2003)
3. Content boosted collaborative filtering (Melville, P., et al, 2002)
4. Content based filtering and collaborative filtering with dynamic user interface (Schafer, J., 2005)

CHAPTER 3. PROPOSED FRAMEWORK

3.1 Data Preprocessing

In this recommendation system, we used the publicly available MoveLens [1] dataset. The MovieLens dataset contains information about movie ratings by users, information about movies and users. The number of movies in the dataset is 1682 and number of users is 943.

We have calculated the average rating, preference and consumption ratio of each user for all the 18 genres. Average rating was normalised for each user. Preference was calculated by multiplying average rating and ratio. Table 3, table 4 and table 5 show normalized average rating, consumption ratio and preference respectively.

Table 1. Normalized Average Rating

user_id	Adventure	War	Drama
1	-0.041737447586281	-0.040610687022900716	0.025543159130945438

Table 2. Consumption Ratio

user_id	Adventure	War	Drama
1	0.27099236641221375	0.15267175572519084	0.04580152671755725

Table 3. Preference

user_id	Adventure	War	Drama
1	-1.131052968941202545e-02	-6.200104888992475673e-03	0.001169915685386814

3.2 Principal Component Analysis (PCA)

Since in our system, the attribute number is really high, before clustering, principal component analysis was used to reduce the dimension from 18 to 5 after analyzing the latent graph in fig 1. Fig.1 shows the variance vs attribute number for average rating data.

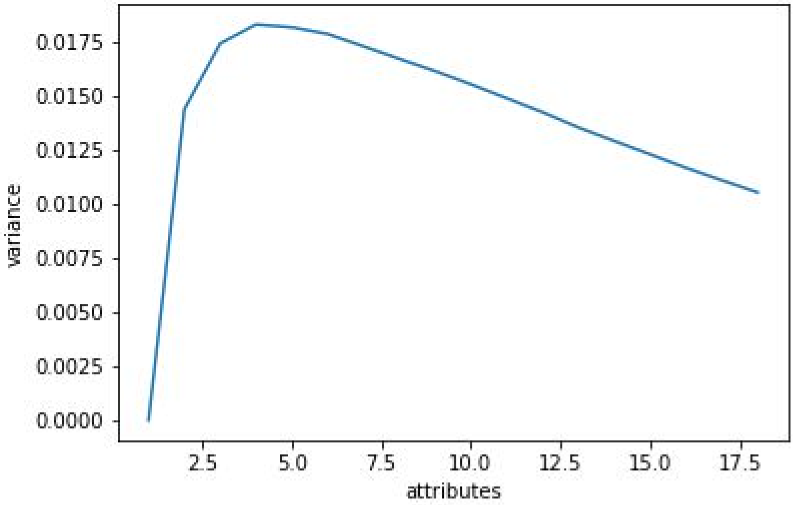


Fig. 2: Variance vs number of attributes

Number of components for dimensionality reduction is chosen as 5 after observing the the variance vs attribute plot.

3.3 Clustering

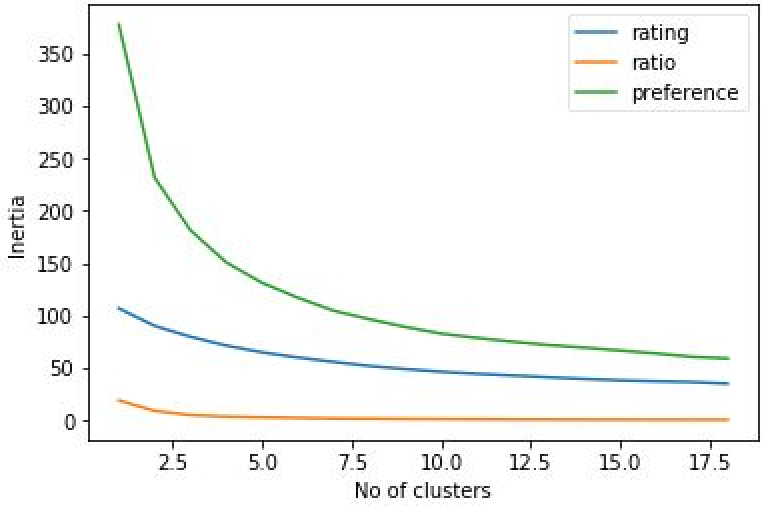


Fig. 3: Inertia vs No. of clusters

Different approaches have been taken to see which approach gives us the best result while clustering the users using k means clustering algorithm. Consumption ratio, user genre preference and user rating has been considered while checking the validity of clustering. Fig.2 shows inertia for each of the approaches.

We decided to go with rating since it has a far superior inertia. Also, after analyzing the cluster number validation graph represented in fig.3, we have decided to go with five clusters judging by elbow method. [elbow]

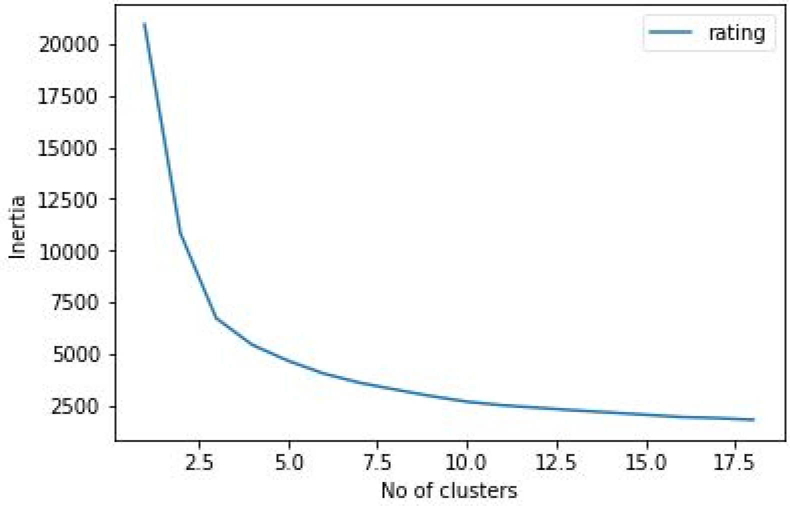


Fig. 4: Cluster number validation

3.4 Data Preprocessing for Neural Network

The dataset contains two files for rating and movie information.

The rating data has the attributes:
User id
Movie id
Rating

The movie data contains genre information of the movies. There were 18 genres.

The rating data and movie data are combined to get rows in the form:

Table 3: Actual movie data

user_id	Action	Adventure	….	Rating
1	1	0	….	4

The rating data and average ratings are multiplied

Table 4: Records after multiplication

user_id	Average Action	Average Adventure	….	Rating
1	-0.041737447586281	0	….	4

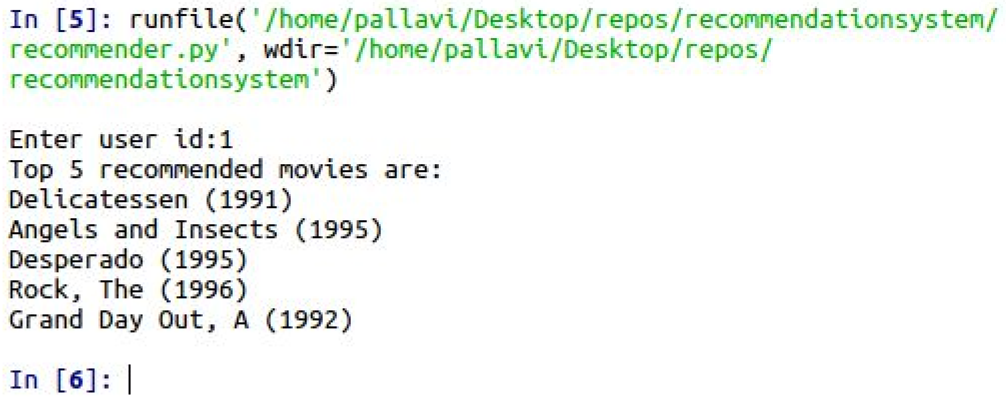
For each cluster, the matrix was split into X containing average rating for the genre information for , i.e. 943x18 matrix and Y containing the ratings, i.e. a 943x1 matrix. This data was fed to the neural network.

3.5 Neural Network

For each cluster 80% of the data was used for training whereas 20% was used for testing. A regressor neural network[34] with 3 hidden layers of sizes 50, 30 and 10. tanh( ) function was used as the activation function.

After the training and predicting the ratings for the movies, the recommendation threshold was set to the average rating given given by each user. If the predicted rating for a given movie is above the average rating of a particular user, the movie is recommended to the user, else the movie is not recommended. The recommended movies are sorted in non-increasing order of predicted rating.

Fig.4 shows the recommendations made for a given user:



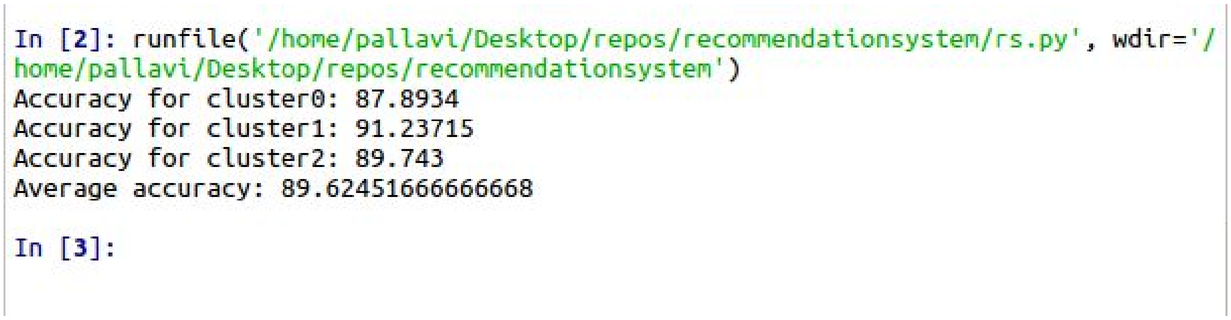
```
In [5]: runfile('/home/pallavi/Desktop/repos/recommendationsystem/
recommender.py', wdir='/home/pallavi/Desktop/repos/
recommendationsystem')

Enter user id:1
Top 5 recommended movies are:
Delicatessen (1991)
Angels and Insects (1995)
Desperado (1995)
Rock, The (1996)
Grand Day Out, A (1992)

In [6]: |
```

Fig. 5: Recommendations for a given user

CHAPTER 4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Results

The result of our system showed on average 89% accuracy depending on the cluster.



```
In [2]: runfile('/home/pallavi/Desktop/repos/recommendationsystem/rs.py', wdir='/
home/pallavi/Desktop/repos/recommendationsystem')
Accuracy for cluster0: 87.8934
Accuracy for cluster1: 91.23715
Accuracy for cluster2: 89.743
Average accuracy: 89.62451666666668

In [3]:
```

Fig. 6: Accuracy for different clusters

CHAPTER 5. CONCLUSIONS

5.1 Summary

We In this paper, we presented different approaches to make movie recommendations. User rating, user consumption ratio and user preference have been considered while building the system. Principal component analysis has been used to reduce dimensionality for better clustering. K-means clustering has been used to group users with similar taste in movies together. Separate neural networks for each cluster have been built to predict rating value of movies given by a user. Our system showed 89% accuracy on average in recommending movies.

5.2 Limitations & Future Work

So far, a lot of research has been done in the field of recommendation systems. But recommendation is not a static problem. The recommendation system uses a static dataset of 943 users and 1682 movies. Efficiency and accuracy can be improved by incorporating further research.

References

1. https://grouplens.org/datasets/movielens/

2.http://citeseer.nj.nec.com/

3.https://en.wikipedia.org/wiki/Recommender_system/

4.https://arxiv.org/ftp/arxiv/papers/1511

5. B. M. Sarwar G. J. A Karypis Konstan J. Riedi "Item-based collaborative filtering recommendation algorithms”.

6 J.S. Breese, D.Heckerman, and C.Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artifical Intelligence, 1998.

7. Martineau, J. C., Cheng, D., & Finin, T. (2013).

8. . Chen, H., Tang, Y., Li, L., Yuan, Y., Li, X., & Tang, Y. (2013).

9. Felden, C., & Chamoni, P. (2007, January).

10. Bouneffouf, D., Bouzeghoub, A., & Gançarski, A. L. (2012).

11. Lucas, J. P., Segrera, S., & Moreno, M. N. (2012

12 Ericson, K., & Pallickara, S. (2011, December).

13. Ericson, K., & Pallickara, S. (2013).

14. Wang, Y., Chan, S. C. F., & Ngai, G. (2012, December).

15. Balabanović, M. (1998). Exploring versus exploiting when learning user models for text recommendation. User Modeling and User-Adapted Interaction

16. Takács, G., Pilászy, I., Németh, B., & Tikk, D. (2009). Scalable collaborative filtering approaches for large recommender systems. The Journal of Machine Learning Research,

17. Borg, M. (2014, September)

18. Shinde, A., Haghnevis, M., Janssen, M. A., Runger, G. C., & Janakiram, M. (2013).

19. . Gorodetsky, V., Samoylov, V., & Serebryakov, S. (2010, August)

20. Hariri, N., Castro-Herrera, C., Mirakhorli, M., Cleland-Huang, J., & Mobasher, B. (2013)

21. Smyth, B., McCarthy, K., Reilly, J., O'Sullivan, D., McGinty, L., & Wilson, D. C. (2005)

22. Alvarez, S. A., Ruiz, C., Kawato, T., & Kogel, W. (2011)

23. . Marović, M., Mihoković, M., Mikša, M., Pribil, S., & Tus, A. (2011, May)

24. Kelley, P. G., Hankes Drielsma, P., Sadeh, N., & Cranor, L. F. (2008, October)

25. Schelter, S., Boden, C., Schenck, M., Alexandrov, A., & Markl, V. (2013, October)

26. Tewari, N. C., Koduvely, H. M., Guha, S., Yadav, A., & David, G. (2013, October)

27. Seric, L., Jukic, M., & Braovic, M. (2013, May)

28. Bhatia, L., & Prasad, S. S. (2015, February)

29. Russell, I., & Markov, Z. (2009, October)

30. Zibriczky, D., Petres, Z., Waszlavik, M., & Tikk, D. (2013, December).

31. Tsapatsoulis, N., Agathokleous, M., Djouvas, C., & Mendez, F. (2015).

32 Yoo, J., Gervasio, M., & Langley, P. (2003, January).

33. Arenas-García, J., Meng, A., Petersen, K. B., Lehn-Schioler, T., Hansen, L. K., & Larsen, J. (2007, August)

34. https://deeplearning4j.org/logistic-regression