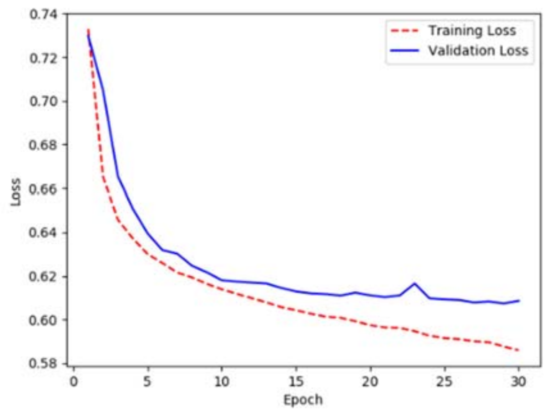

Fig 1. DeepAutoEnCF-U Training Vs. Validation Loss for MovieLens-1M dataset (80/10/10 split) Without Features

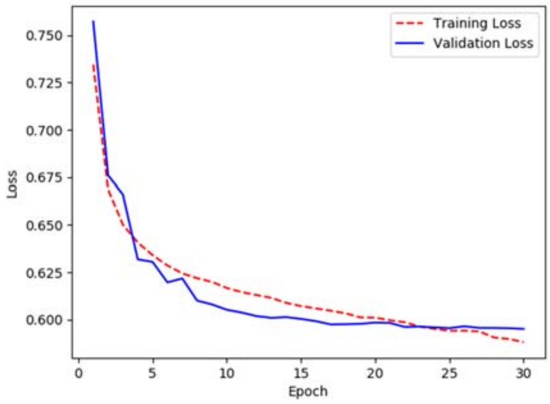

Fig 2. DeepAutoEnCF-U Training Vs. Validation Loss for MovieLens-1M dataset (80/10/10 split) with Age and Gender Features

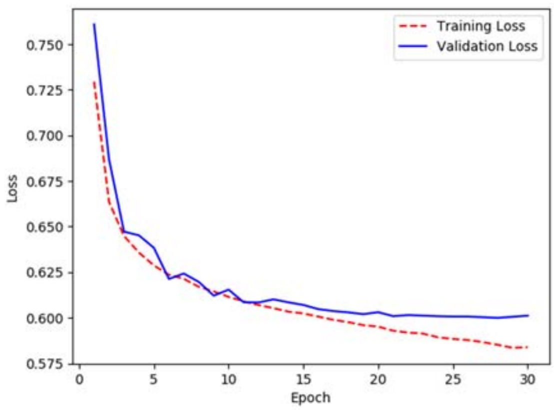

Fig 3. DeepAutoEnCF-U Training Vs. Validation Loss for MovieLens-1M dataset (80/10/10 split) with Age, Gender and Movie Genre Weightage Features

Fig. 4 shows performance of DeepAutoEnCF-U on MovieLens-1M dataset for 60/20/20 split with features. The model overfits for this split with features. It shows similar performance with and without features for this split.

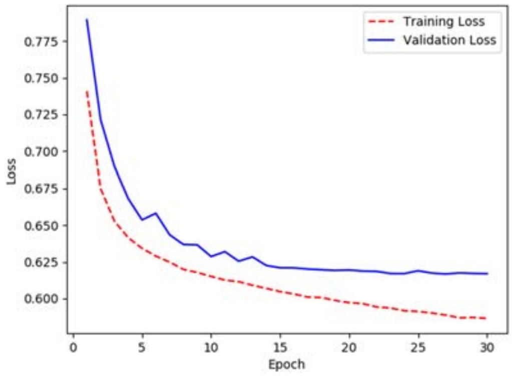

Fig 4. DeepAutoEnCF-U Training Vs. Validation Loss for MovieLens-1M dataset (60/20/20 split) with Age, Gender and Movie Genre Weightage Features

Fig. 5 shows the training loss against the validation loss of DeepAutoEnCF-I for MovieLens-1M dataset without any features. Fig. 6 shows the training loss against the validation loss of DeepAutoEnCF-I on MovieLens-1M dataset with movie genre feature. Here also the model performs better with features which can perform better when the model is generalized.

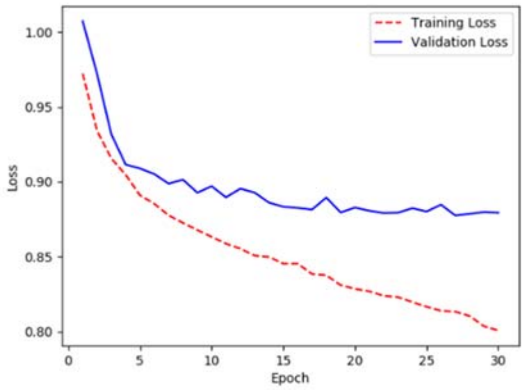

Fig 5. DeepAutoEnCF-I Training Vs. Validation Loss for MovieLens-1M dataset (80/10/10 split) without Features

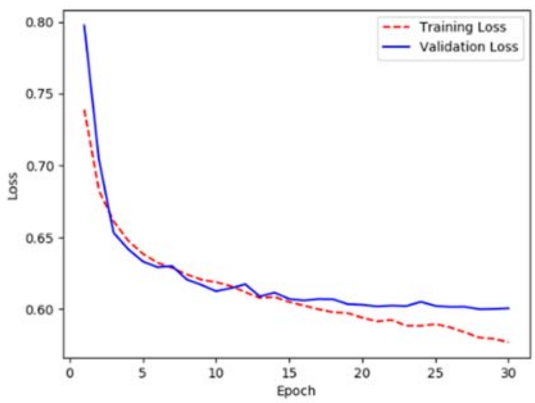

Fig 6. DeepAutoEnCF-I Training Vs. Validation Loss for MovieLens-1M Dataset (80/10/10 split) with Movie Genre Feature

Fig. 7 shows performance of DeepAutoEnCF-I for MovieLen-1M dataset for 60/20/20 dataset split. For item based approach also the model shows either overfit or unpresentable performance. For both the approaches the performance (RMSE) and the learning improves when features are added to the input with 90/10/10 dataset split.

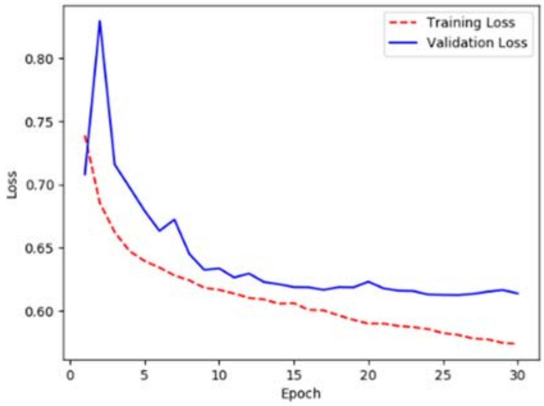

Fig 7. DeepAutoEnCF-I Training Vs. Validation Loss for MovieLens-1M Dataset (60/20/20 split) with Movie Genre Feature

7. Conclusions

For both the datasets, the proposed system outperforms the traditional collaborative filtering algorithms. For MovieLens 100K dataset DeepAutoEnCF-I showed better performance than DeepAutoEnCF-U. For MovieLens-1M dataset both the approaches, user based and item based, show improvement in performance when features are added to the input. The performance of DeepAutoEnCF-I improved more than DeepAutoEnCF-U as compared to their performance without features. Whenever the data is spare, adding new features supports the learning process and improves the performance. Denoising Autoencoders with dropout improve the performance of recommender systems for predicting ratings. New features can be thought and added to the input further for improving the performance of recommender systems with deep learning through denoising autoencoders.

References

[1] Betru B. T., Onana C. A. and Batchakui B. (2017), "Deep Learning Methods on Recommender System: A Survey of State-of-the-Art", International Journal of Computer Applications, Volume 162 – No – 10, March 2017.
[2] Dong X.,et al. (2017), "A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems", Thirty first AAAI Conference on Artificial Intelligence.
[3] Ekstrand M. D., Riedl J. T. and Konstan J. A. (2018), "Collaborative Filtering Recommender Systems", Foundations and Trends in Human–Computer Interaction Vol. 4, No. 2, 81–173.
[4] Goodfellow I., Bengio Y. and Courville A. (2016), "Deep Learning", MIT Press.
[5] Harper M. and Kostan J. A. (2015), "The MovieLens Datasets: History and Context", ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19, 19 pages. DOI = http://dx.doi.org/10.1145/2827872
[6] Isinkaye F. O., Folajimi Y. O. and Ojokoh B. A. (2015), "Recommendation systems: Principles, methods and evaluation", Egyptian Informatics Journal 16, 261-273.
[7] Koren Y. (2009), "Matrix Factorization Techniques for Recommender Systems", IEEE Computer Society.
[8] Kuchaiev O. and Ginsburg B. (2017), "Training Deep AutoEncoders for Collaborative Filtering", arXiv.
[9] Li S., Kawale J. and Fu Y. (2015), "Deep Collaborative Filtering via Marginalized Denoising Auto-encoder", CIKM, ACM.
[10] Liang J. and Liu R. (2015), "Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network", 8th International Congress on Image and Signal Processing (CISP).
[11] Sarwar B., et al. (2001), "Item-Based Collaborative Filtering Recommendation Algorithms", WWW10, ACM.
[12] Sedhain S., et al. (2015), "AutoRec: AutoEncoders meet collaborative filtering", WWW, ACM.
[13] Srivastava N., et al. (2014), "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", Journal of Machine Learning Research 15, 1929-1958.
[14] Strub F., Gaudel R. and Jeremie M. (2016), "Hybrid Recommender System Based on Autoencoders", Workshop on Deep Learning for Recommender Systems, Boston US, ACM.
[15] Su X. and Khoshgoftaar T. M. (2009), "A Survey of Collaborative Filtering Techniques", Advances in Artificial Intelligence.
[16] Vincent P., et al. (2010), "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", Journal of Machine Learning Research 11, 3371-3408.
[17] Volkovs M., Yu G. and Poutanen T. (2017), "DropoutNet: Addressing Cold Start in Recommender Systems", NIPS.
[18] Wu Y., et al. (2016), "Collaborative Denoising Auto-Encoders for Top-N Recommender Systems", WSDM, ACM.
[19] Zhang S., et al. (2017), "Hybrid Collaborative Recommendation via Semi-AutoEncoder", 4th International Conference, ICONIP.
[20] Zhang S., Yao L. and Sun A. (2017), "Deep Learning Based Recommender System: A Survey and New Perspectives", ACM J. Comput. Cult. Herit., Vol. 1, No. 1, Article 35.