



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

PERSONALIZING ACADEMIA USING DEEP NEURAL NETWORKS

¹S Sree Harsha, ²Kotian Prajwal Vijay, ³Swasthik BS, ⁴Dr. Rekha Phadke

¹Student, ²Student, ³Student, ⁴Asst. Professor

¹Dept. of Electronics & Communication,

¹Nitte Meenakshi Institute of Technology, Bangalore, India

Abstract: As students progress through their higher education, they find themselves in an Environment saturated with academic content available both offline in the form of subjects they can choose from or textbooks they can refer to and online in the form of educational content presented on various online platforms. Every student has a different method of understanding and getting stuck with a wrong reference can cause a slump in that topic, the worst-case scenario choosing a subject under pretences may lead to failure. The system itself has birthed the need for a platform that can give the students, not just accurate information pertaining to all the choices in front of them but also a personalized learning guide. This is what we strive to solve in our project.

Index Terms – Recommender Systems, Deep Neural Networks, Collaborative Filtering, Matrix Factorization, Multi-layered perceptron

I. INTRODUCTION

A Recommender system makes use of User data such as their ratings on existing items and other information to predict recommendations for the users that they might like in the future. This system sits at the core of various services such as online retail services, streaming services, and even social networks. Our goal is to create an all-inclusive platform that can give well-tailored recommendations based on the user's profile, made using DNN (Deep Neural Network) that would classify the content available and learn the client's preferences in real-time using their history and other metadata. The platform will contain a section with detailed descriptions of subjects taken from primary sources, approaches that can be taken to tackle it as well as recommendations for authors and content creators to help improve understanding. The deep learning algorithm will be divided into two sections:

1. Classification: We aim to create a huge Dataset with resources curated by college faculties, local and international authors as well as external sources like Coursera and YouTube. The algorithm will classify this content into well-defined categories such as theory, equation-heavy, visualization.

2. Ranking: The algorithm will consequently rank the material present inside each category based on user interactions, and previous history. This would generate the best content based on user preferences.

We have implemented our project as a fully functional website with multiple pages, using Web Framework called Flask.

II. OBJECTIVES

The following are the objectives of present work:

- The Aim is to determine a feasible method to implement recommender systems in the academic field.
- To learn about the preferences and behaviours of existing users to make the experience better for new users.
- To collect accurate information from various sources and make it available for the user regarding subject choices with testimonies from previous users.
- To collect and analyse the data from various websites, educators, authors and provide a search engine service.

III. METHODOLOGY AND IMPLEMENTATION

3.1 Methodology

Figure 3.1 illustrates the functional flow process of generating recommendations and the inputs and resources involved.

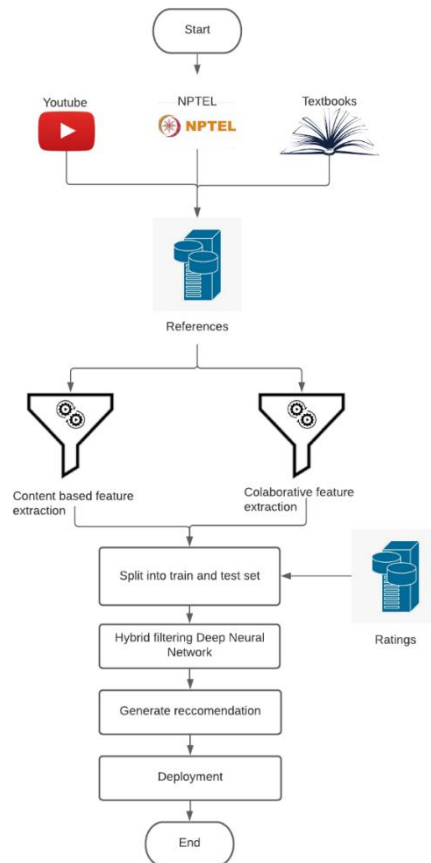


Figure 3.1: Functional Diagram of Recommender System

3.2 Implementation

This project can be divided into three major sections:

3.2.1 Dataset Creation

Deep Learning models by nature are data hungry. Recommendation systems have been exclusively used by companies that have a mammoth share of the market like Netflix and Youtube because of their ability to acquire massive amounts of data. An institute of a smaller scale cannot hope to garner such information from a limited user base. The resource crunch presented here was solved by creating hypothetical users who amplified the tastes of the user reviews that we found attached to the academic materials online.

We have used a custom made Dataset for the purpose of our project, and will not be releasing it as a part of this paper. The project dataset consists of two datasets. References dataset is a collection of academic material that were sourced from commonly used resources by students like Youtube, NPTEL, and textbooks available on a particular subject. The data indicates the author, the date of publishing, and an internal metric of approach where we labelled academic material according to their approach of teaching. The labels being Mathematical(M), Theoretical (T), and Visual (V). These labels were sourced from screening each material in the dataset. Real-world user reviews were considered while drafting the references. Such data gives ideal features for recommender systems.

```
references.head()
```

Item ID	Semester	Title	Author	Year	Origin	Approach	Subject	
0	1	7	Optical Fiber	John Gowar	2001	F	T	Optical Fiber Communication
1	2	7	Optical Fiber Communications	Gerd Keiser	2015	F	M	Optical Fiber Communication
2	3	7	Power Electronics, Circuits, Devices and Appli...	M.H.Rashid	2002	L	T	Power Electronics
3	4	7	Power Electronics	M D Singh, K B Khanchandani	2009	L	M	Power Electronics
4	5	7	Power Electronics (https://onlinecourses.nptel...	Prof. G.Bhuvaneshwari	2018	L	V	Power Electronics

Table 3.2.1: References Dataset

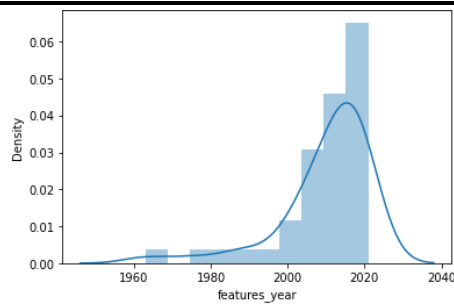


Figure 3.2.1: Distribution of references with respect to years.

An ideal feature for a recommender system is one that can either define the nature of the content or determine user behaviour. Content-based filtering and Collaborative filtering are the two approaches taken to operate on the features respectively. In our case we decided to choose two features from References dataset namely Approach and Year of publication, each due to their ability to characterise the material. These features will be further extracted with Content-based filtering. Ratings dataset contains the ratings given by pre-existing users to the academic material. These ratings represent a dynamic feature which determine, on a user-to-user basis, the tastes of individuals. These features rather than characterising the material, characterise the user and thus need a different filtering technique called Collaborative filtering. To resolve the lack of access to a mass audience, we used pandas library from python3.8 to amplify the ratings given to these academic materials online.

The target audience chosen for this project were students of Electronics and Communications engineering from 3rd semester to the final semester. Each subject on an average has 20 materials, each of which has been taken from VTU and college provided recommendations and from personal experience. All the materials are screened individually to determine to quality based on online reviews and these reviews are then expanded upon by using pandas and are added to the two datasets.

```
rating.head(10)
```

	User ID	Item ID	Rating
0	1	7	9
1	1	30	2
2	1	1	7
3	1	17	6
4	1	27	1

Table 3.2.2: Ratings Dataset

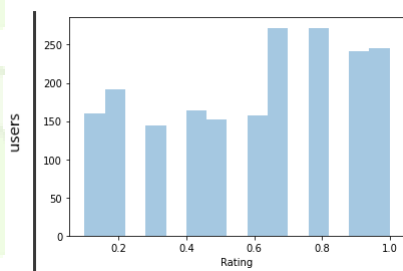


Figure. 3.2.2: Distribution of Ratings w.r.t users

3.2.2 Deep Learning Model

Recommendation generation is not a typical deep learning application. There are multiple ways to approach the problem Content-based filtering, Collaborative filtering, Matrix factorization, SVD, Hybrid, etc. The decision between the methods is made based on the type of the problem. These methods can be executed using traditional Machine learning and do not require the interjection of deep learning unless the problem is non-traditional. For this project, the problem included sparse data and thus we decided to take a Hybrid approach.

This Hybrid Deep learning model had two components; Matrix factorization and a Multi layered perceptron. Matrix factorization is applied to generate latent features and MLP is used for thorough comprehension of the features. Used in conjunction these topologies provide for a fairly strong generalization of features methodology.

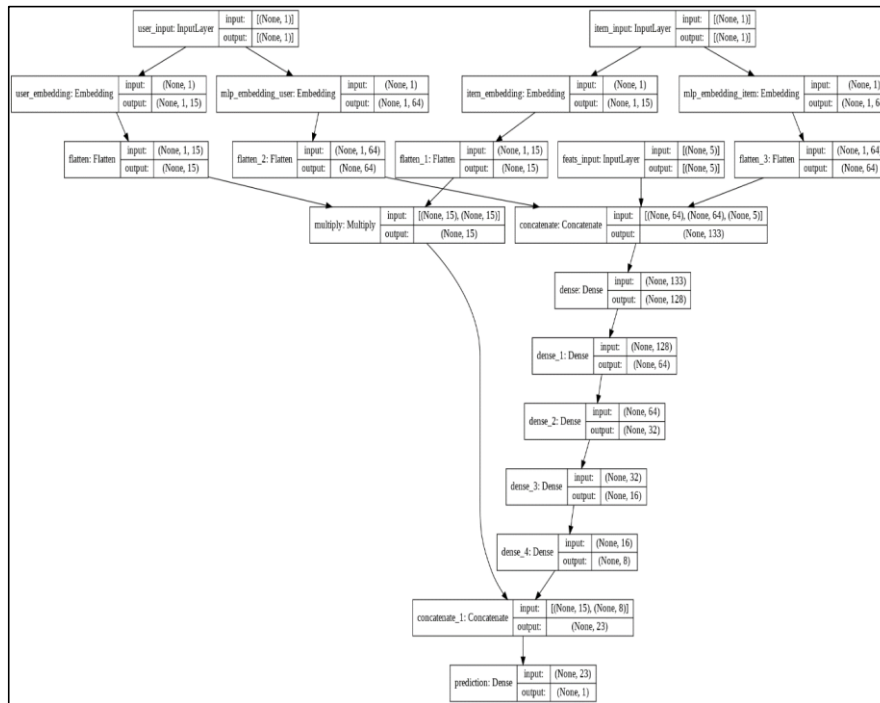


Figure. 3.2.3: Model Architecture

For the sake of pre-processing of data, we are first one hot encoding the approach feature. This is done to create a feature matrix which can be defined as a finite set of properties with respect to which a given set of linguistic units can be characterized by a set of features. This feature matrix is the input required for matrix factorization where latent features are then extracted. We take two initial inputs with user-based embeddings and item-based embeddings. We take two of each input so that we can feed them into both MLP and Matrix factorization. The layers then flatten the features and multiply them for matrix factorization and for MLP they concatenate the inputs and keep passing them through dense layers. The two paths are later concatenated again and later passed through a single filter output layer.

The output layer is tasked with taking the features and providing a score to all the materials. This hierarchy of scores sorted in descending order indicate the system’s recommendations for a specific user. These scores are provided to all the materials present in the dataset and are further sorted by semester and subject for user convenience. The loss function that we use for training is Mean square logarithmic error, which as the name suggests is a variation of Mean Squared error. MSLE is preferred for a recommendation task as it only considers percentual differences in the prediction for e.g.: if a prediction rating is 9 but the user rated it as 8, the error rate will be less than if the user rating would have been 2. This allows a balanced view of the errors made by the model.

True value	Predicted value	MSE loss	MSLE loss
30	20	100	0.02861
30000	20000	100 000 000	0.03100
	Comment	big difference	small difference

Table 3.2.3: MSLE loss example

3.2.3 Flask Web Framework:

To be able to deploy our project and provide the end user with a working service, we used Flask Web Framework to build and configure webpages. Flask is called a micro-framework that provides just the essential features for a webpage, such as creating and handling sessions, request handling and routing webpages, as well as extra features to use python and HTML together.

Therefore, Flask Framework can handle front-end and back-end programming. It uses a templating engine called Jinja2, which serves as a programming language for writing scripts within HTML code, to make the webpage more dynamic without the need of additional JavaScript.

IV. RESULTS AND ANALYSIS

4.1 Model Performance

The result of the model is as follows:

Epochs: 50 Epochs

Training Accuracy: 80.56%

Validation Accuracy : 75.94%

Training loss: 0.0015

Validation loss: 0.0441

```
Epoch 49/50
31/31 [=====] - 0s 5ms/step - loss: 0.0015 - accuracy: 0.8056 - val_loss: 0.0446 - val_accuracy: 0.7571
Epoch 50/50
31/31 [=====] - 0s 5ms/step - loss: 0.0015 - accuracy: 0.8056 - val_loss: 0.0441 - val_accuracy: 0.7594
```

Figure 4.1: Training Output

4.2 WebApp Interface

The Recommendations are obtained from the Model and the Top 4 predictions are displayed in the webpage.

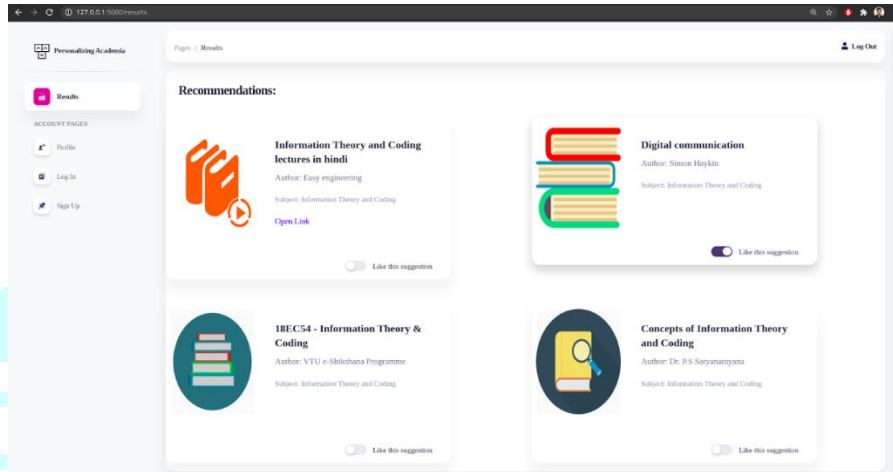


Figure 4.2: Results Page

Features on the Results Page:

- Choice to give Feedback on each recommendation, which will be used to update the ratings data and help fine tune the predictions later.
- If the recommendation is a YouTube video or NPTEL course material, then a Link for the same will be added.
- All the other subject material that did not make it to the Top 4 Recommendations will be shown at the bottom of the page.

V. CONCLUSION

We conclude that, by utilising a Hybrid approach to a Deep Neural Network, a Recommender System can be created even with sparse data. Using proper pre-processing techniques, we can optimize available data to extract latent features, which help us understand a User's subjective needs.

For the future scope of a similar venture, the question of how to further optimize sparse data inputs and implement an identical system for a smaller scale institution, can be addressed. Furthermore, latent feature extraction on a wider spectrum of available features can also be achieved. In this project we have tried to democratize the use of Recommender Systems by achieving high efficiency with sparse data, which signifies a shift of this technique catered towards small-scale players in the market, with minimal infrastructure cost. The custom-made Webpages and Dataset used in this project will not be released as a part of this paper.

REFERENCES

- [1] *Building Recommender Systems with Machine Learning and AI: Help People Discover New Products and Content with Deep Learning, Neural Networks, and Machine Learning Recommendations*
- [2] Z. Chen, W. Liang, X. Gao, Z. Zhou and M. Wu, "Research on the Accurate Recommendation Management System for Employment of College Graduates on Hadoop," 2019 5th International Conference on Big Data and Information Analytics (BigDIA), Kunming, China, 2019, pp. 19-22, doi: 10.1109/BigDIA.2019.8802855.
- [3] R. Obeidat, R. Duwairi and A. Al-Aiad, "A Collaborative Recommendation System for Online Courses Recommendations," 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), Istanbul, Turkey, 2019, pp. 49-54, doi: 10.1109/Deep-ML.2019.00018.
- [4] W. Tanner, E. Akbas and M. Hasan, "Paper Recommendation Based on Citation Relation," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 3053-3059, doi: 10.1109/BigData47090.2019.9006200.
- [5] S. Chaudhari, A. Sawant and R. Patil, "Self-tuning approach for implementing a multidimensional recommendation system using PID," 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 2017, pp. 1-4, doi: 10.1109/IPACT.2017.8244895.
- [6] J. Wei, J. He, K. Chen, Y. Zhou and Z. Tang, "Collaborative Filtering and Deep Learning Based Hybrid Recommendation for Cold Start Problem," 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, Auckland, New Zealand, 2016, pp. 874-877, doi: 10.1109/DASC-PICOM-DataCom-CyberSciTec.2016.149.
- [7] M. Inoue, H. Takenouchi and M. Tokumaru, "Music Recommendation System Improvement Using Distributed Genetic Algorithm," 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS), Sapporo, Japan, 2016, pp. 627-630, doi: 10.1109/SCIS-ISIS.2016.0136.
- [8] Xuesong Zhao and Kaifan Ji, "Tourism e-commerce recommender system based on web data mining," 2013 8th International Conference on Computer Science & Education, Colombo, Sri Lanka, 2013, pp. 1485-1488, doi: 10.1109/ICCSE.2013.6554161.
- [9] J. Sun and Y. Xie, "A Web Data Mining Framework for E-Commerce Recommender Systems," 2009 International Conference on Computational Intelligence and Software Engineering, Wuhan, China, 2009, pp. 1-4, doi: 10.1109/CISE.2009.5363548.
- [10] C. Haruechaiyasak, M. -. Shyu and S. -. Chen, "A data mining framework for building a Web-page recommender system," Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004., Las Vegas, NV, USA, 2004, pp. 357-362, doi: 10.1109/IRI.2004.1431487.
- [11] Flask Web Framework - <https://flask.palletsprojects.com/>

