A MAJOR PROJECT II REPORT

ON

EMPLOYING AUTOENCODER IN BUILDING A MOVIE RECOMMENDATION SYSTEM

Submitted in Partial Fulfillment of the Requirements

for the Award of the Degree of

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:

ANURAG KUMAR CHAUHAN

Roll No-2K22/CSE/05

Under the Supervision of

Dr. VINOD KUMAR

(HOD, Department of Computer Science & Engineering)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042. India

MAY,2024

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE DECLARATION

I **ANURAG KUMAR CHAUHAN** Roll number 2K22/CSE/05 student of M.Tech (Computer Science and Engineering), hereby declare that the work which is being presented in the major project - II entitled "**Employing Autoencoders in Building a Movie Recommendation System"** which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **Anurag Kumar Chauhan**

Date:

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE

I hereby certify that the Major Project Report II titled **"Employing Autoencoders in Building a Movie Recommendation System"** which is submitted by **ANURAG Kumar Chauhan**, Roll No 2K22/CSE/05, Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date:

Dr. VINOD KUMAR

(SUPERVISOR)

Head of Department,

Department of CSE,

DTU, Delhi

ACKNOWLEDGEMENT

I would like to express my deep appreciation to **Dr. Vinod Kumar**, Head of department at the Department of Computer Science and Engineering, Delhi Technological University, for his invaluable guidance and unwavering encouragement throughout the course of this research. His vast knowledge, motivation, expertise, and insightful feedback have been instrumental in every aspect of preparing this research plan. His expertise and scholarly guidance have significantly enhanced the quality of this thesis.

My heartfelt thanks go out to the esteemed faculty members of the Department of Computer Science & Engineering at Delhi Technological University. I extend my gratitude to my colleagues and friends for their unwavering support and encouragement during this challenging journey. Their intellectual exchanges, constructive critiques, and camaraderie have enriched my research experience and made it truly fulfilling.

While it is impossible to name everyone individually, I want to acknowledge the collective efforts and contributions of all those who have been part of this journey. Their constant love, encouragement, and support have been indispensable in completing this MTech report.

Anurag Kumar Chauhan
(2K22/CSE/05)

ABSTRACT

We made a system that gives you suggestions based on what you like using a type of neural network. We use a model that has two parts to make suggestions based on what you like. Autoencoders are a kind of AI that can be used to make movie suggestions. This paper is about how to make computer systems that can recommend movies to you by themselves. We also talk about different kinds of autoencoders, like ones that remove noise, ones that keep only important parts, ones that learn to compress data, and ones that stack multiple autoencoders. The paper also looks at the good, bad and real-life stuff of using autoencoders for movie suggestions.

This means that having lots of movie data with what people like and what the movies have is really helpful for autoencoders to learn from it. This paper tries to make movie recommendation systems better by using Autoencoders and testing different ways of doing things with them. Checking out the results from experiments on a public database shows how much better the usual ways of doing things are.

In short, these systems can help you find movies that you like by using your own preferences and feedback.

LIST OF PUBLICATIONS

1. Anurag Kumar Chauhan, “A Review on Recommendation Systems Utilizing Diverse Collaborative Filtering Algorithms”, Accepted at **“International Conference on Intelligent Computing and Communication Techniques (ICICCT)”**, at JNU New Delhi, India.

   Paper Id: 672

   Indexed by Scopus.

2. Anurag Kumar Chauhan, “Employing Autoencoders in Building a Movie Recommendation System” Accepted at **“National Conference on Advanced Computer Science and Information Technology (NCACSI - 24)”,** Udaipur, India.

   Paper Id: National Conference_8175362

   Indexed by Scopus.

TABLE OF CONTENTS

List of Tables

List of Figures

LIST OF ABBREVIATIONS

- **RS** Recommendation System
- **AE** Autoencoder
- **DAE** Denoising Autoencoder
- **VAE** Variational Autoencoder
- **SAE** Stacked Autoencoder
- **DNN** Deep Neural Network
- **ANN** Artificial Neural Network
- **AF** Activation Function

CHAPTER 1

INTRODUCTION

1.1 Overview

The system looks at what the user has chosen to make a decision. It's an essential element that propels deeper client connection, tailored user experiences, and potent decision support tools in the retail, entertainment, healthcare, and financial sectors. A products rating or preference is estimated by recommendation systems. Neural network-based techniques are widely used in recommender systems. Recommendation systems are used to suggest movies, music, and videos in the entertainment industry. The most popular form of entertainment is movies. Video content suggestions from YouTube and Amazon Prime are determined by the same set of standards. Due to the vast amount of material available on the internet, it can be difficult to discover your content. Many people are starting to use suggestions to filter out information in various forms of media, such as books, movies, music, videos, and social media. Collaborative filtering is used the most by businesses due to its increased emphasis on user engagement. Collaborative filters analyse a user's browsing history, compare it with other user's histories, and the results are more likely to be precise if they are based on both the content of the items and other factors. Movies that have similar content to what the users like are found by using cosine similarity. and then recommends them from most to least relevant based on the user's input. Context-based filtering utilizes user-provided information such as genre release date and mood to enhance the accuracy of its outcomes. There are multiple methods for suggesting movies.

1.2 Types of RS

1.2.1 Content based filters

The recommendation system requires information on what the user likes or dislikes. It's based on previous action or feedback. The tags and genre serve as a foundation for content-based filters. This approach combines different genres and tags and produces the most accurate results. A person is more likely to see the most watched genre in the descending order if they watch the education documentary genre more than the action genre.

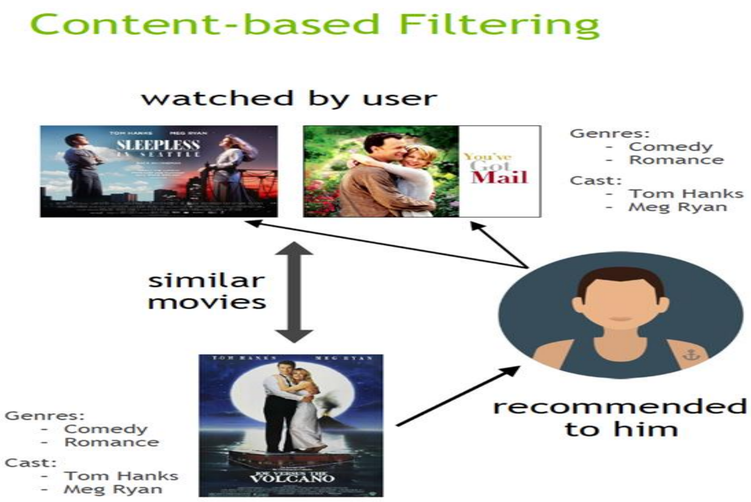


Figure 1.1 Content based filtering

1.2.2 Collaborative based RS

One of the methods employed to produce recommendations in the recommendation system is Collaborative filtering. This approach looks for user similarities. The method in the system that recommends products by looking at how users who like the same things are related is called collaborative filtering. the approach aims to predict future events by identifying similarities among users. For example, if two users, A and B, share similar characteristics and A has shown interest in product A, it is reasonable to assume that B will also be interested in product A. Consequently, B can be recommended to try product A. Collaborative filtering involves gathering user opinion data about commodities and using that information to determine the environment based on user similarities.

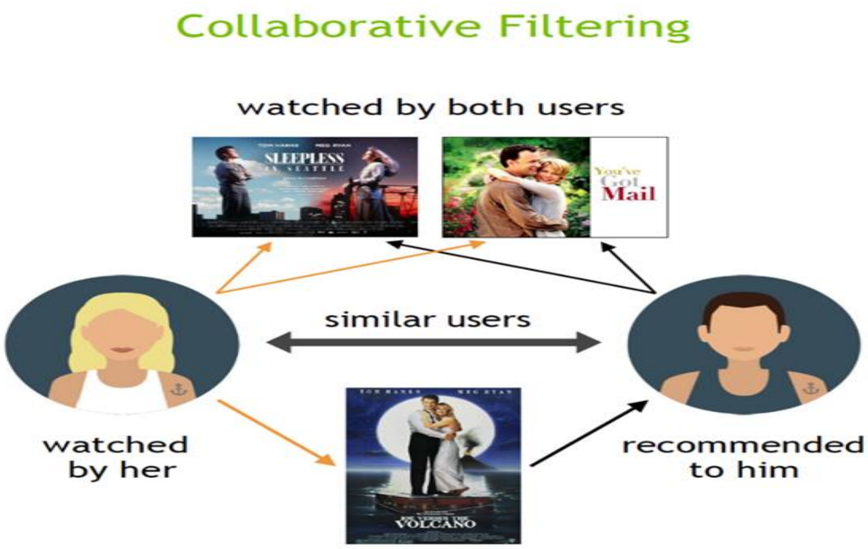


Figure 1.2 Collaborative filtering

1.2.3 Hybrid RS

Clients are provided with a broader selection of products by employing collaborative and content-based filters. the new recommender system is believed to be more precise than other systems. This is a remarkable illustration of a mixed-genre suggestion algorithm. Can you rephrase this sentence: It examines the behavior of users to identify individuals with similar interests on that platform. This enables it to provide suggestions. This is the method Netflix employs for collaborative filtering. The system recommends content that is similar to the ones that the user has given high ratings for. the frequent problems with recommendation systems include issues with cold starts and insufficient data.

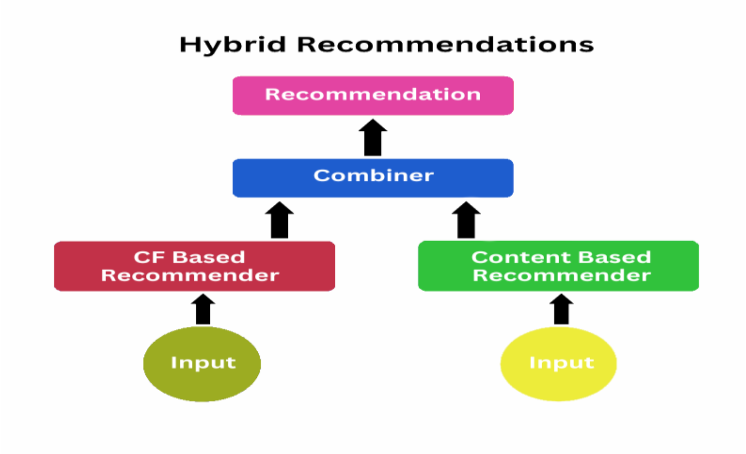


Figure 1.3 Hybrid recommendation

1.2.4 Context-Aware RS

The variables that the recommender systems take into account are time, location, and device. As an example, a music recommendation system could take into account the users geographical location and the current time to propose suitable music for the specific event.

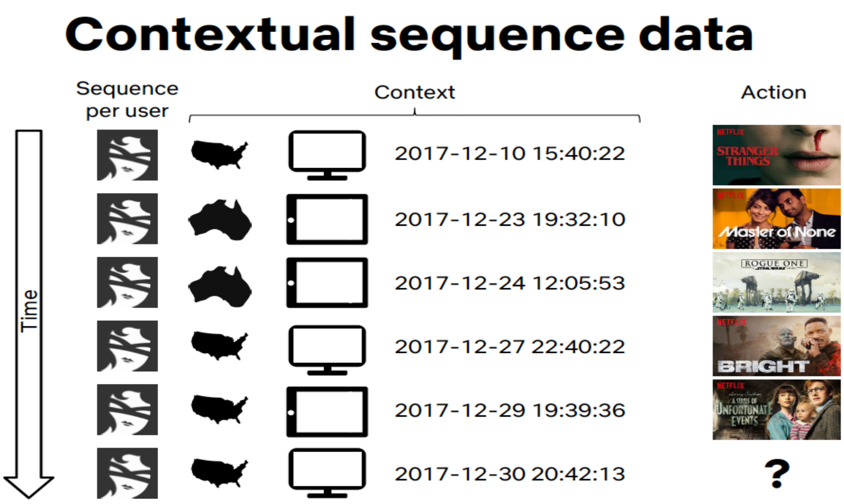


Figure 1.4 context aware recommendation system

1.2.5 The Deep NN RS

Deep neural network recommendation systems can predict user preferences. The systems can comprehend intricate patterns and connections in the data by applying deep learning techniques. These systems can learn complicated patterns from data because they have many networks that are connected to each other. By means of ongoing training and improvement, these systems adjust to changing user preferences and behaviours, offering tailored recommendations across a range of industries.

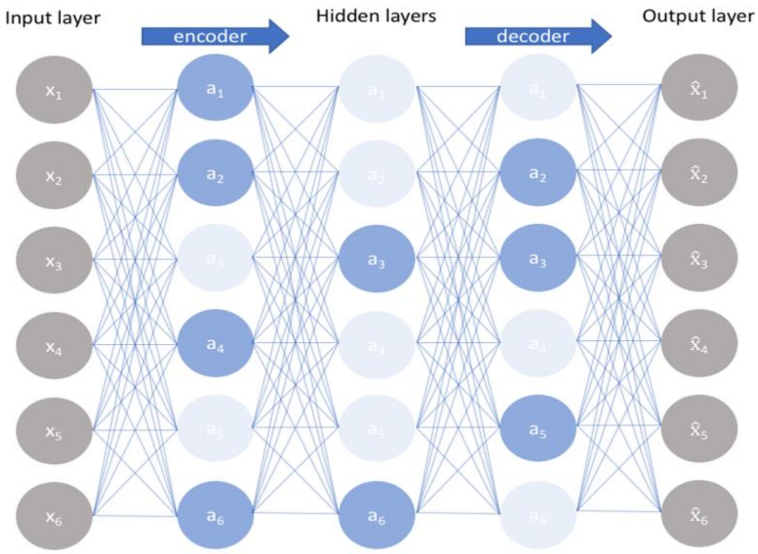


Figure 1.5 deep neural network based recommendation system

1.3 Project Objectives

Were using a fancy machine learning thingy to make a model that can give you movie ideas that match your taste. We want to make a thing that can handle a lot of data and make people happy. The goal of this project is to collect and get ready user ratings, make and teach a machine to make small versions of users and movies, and add the system to a simple and user-friendly app. The systems efficiency will be checked against regular collaborative filters, and the systems effectiveness will be measured using specific metrics. The main objective is to create a strong recommendation system that learns from new user data, leading to an enhanced movie-watching experience for users.

The project's primary goal is:

- To make the architecture of stacked autoencoder.
- It's important to choose a configuration that gives the best results.
- Train the suggested system.
    - The goal of training the model is to reduce the loss.
- Testing the proposed system
    - To enhance the efficiency of the model, it is tested for different epochs.

1.4 AutoEncoder

The Autoencoder is a machine learning technique that learns the features of data by reconstructing it using two processes. The AE has three layers, one that gets the input, one that does something with it and one that gives the result. The hidden layer is needed to make the input data look like the output data so that we can measure how much they differ.

There are two processes of AE comprise the following

1. The transformation of input data from the initial layer to the hidden layer is illustrated in equation(1):

$$Z = f(wX + b) \qquad (1)$$

2. The procedure of converting from the hidden layer to the output layer is illustrated in equation(2):

$$Y = g(w'Z \ + b') \qquad (2)$$

Where w and w' is weight b and b' is bias used in AE neural network. X is input matrix, Y is output representation of input matrix and Z is hidden layer representation of input matrix. f() and g() is activation function.

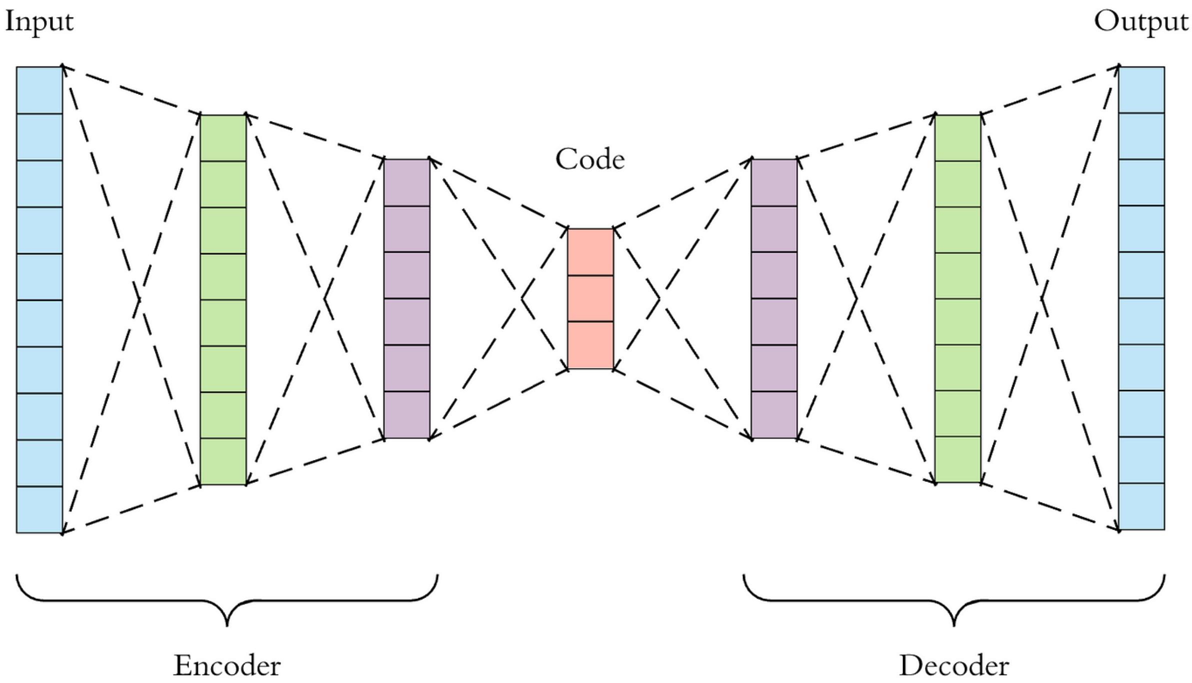


Figure 1.6 Autoencoder

AE contains following components:

- **Encoder**: It reduces the input data into a smaller size representation. The model just takes a short version of the input, called the "bottleneck," which is made by putting some modules together or using some simple lines.
- **Code**: It is the most important part of AE. It extracts the minimum feature from the input to rebuild output. The compressed code causes the neural network to struggle with memorizing the input and overfitting the data. The chance of overfitting is reduced with the smaller code. This layer can be used as a linear layer or a tensor layer.
- **Decoder**: The network's decoding component acts as an interpreter for the code. The network can be degraded by the decoding of the knowledge representations. The ground truth is compared to the output. If we use images or linear layers, this is usually implemented with transposed convolutions.

The introduction of autoencoders created new possibilities for recommender systems to be reinvented in order to meet customer needs:

- Unlike traditional models that focus on a single data source, auto-encoder based models can handle multiple data sources.
- The improved recommendation accuracy is a result of the enhanced comprehension of user needs and item characteristics that auto-encoder possesses.
- The model proposed by the author is capable of handling data with noise better than the conventional models, thanks to its auto-encoder component.

1.5 Types of AE

1.5.1 Denoising AE

The input data is cleaned by a denoising autoencoder (DAE) which is a kind of autoencoder. The input data is modified to make it look different, and the model learns from these variations. The autoencoder can be used to eliminate noise in new, unfamiliar data by learning to recreate the original input data from the corrupted version.

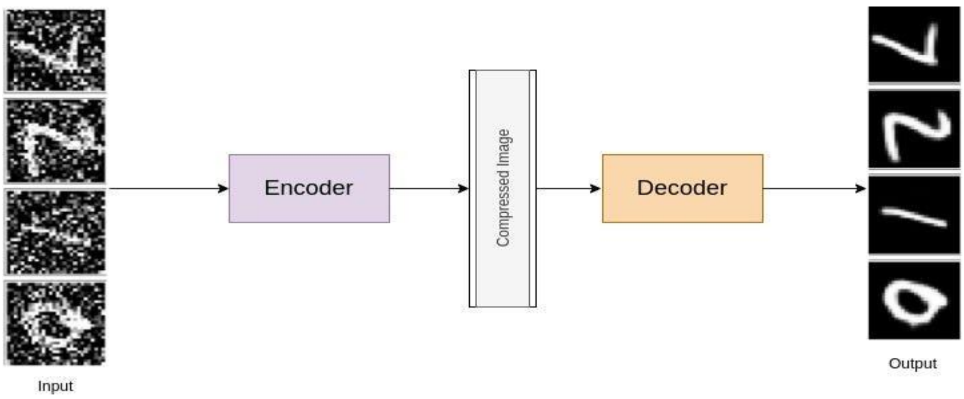


Figure 1.7 Denoising autoencoder

1.5.2 Sparse AE

The sentence can be paraphrased as: A sparse autoencoder is a type of autoencoder that reduces the size of the input data and makes the output data have few non-zero values. The sparsity constraint motivates the autoencoder to learn a representation with a limited number of neurons. To make the hidden layer activations more similar to a small target value, we can use a regularization term in the loss function that penalizes the activations that are too large or too small.

A sparse autoencoder includes a penalty for the sparsity of its output. In most instances, we would construct our loss function by imposing penalties on the activations of hidden layers, thereby encouraging only a select few to utilize the network.

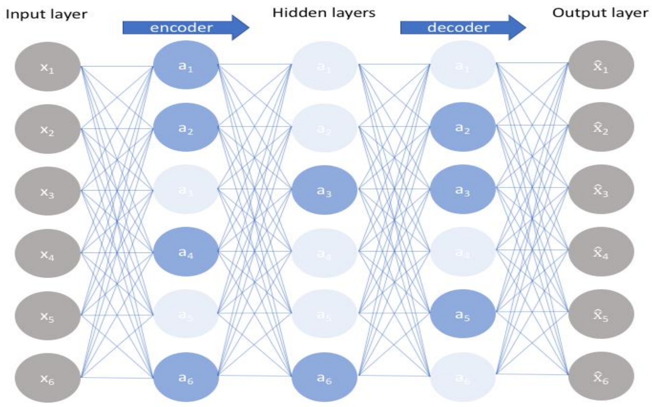


Figure 1.8 Sparse autoencoder

1.5.3 Variational AE

The underlying probability distribution of a given dataset can be captured by generative models, which can then be used to create new samples. They utilize a structure that includes an Encoder-decoder component. The data is transformed into an unknown form and the data is reconstructed using this representation. The VAE is designed to reduce the disparities between the original and reconstructed data, enabling it to understand the underlying data distribution and produce new samples that align with the same distribution.

VAEs can create new data that look like the data they learned from. The VAE can generate new data points by using a continuous space. It can also estimate the density of data and generate text.

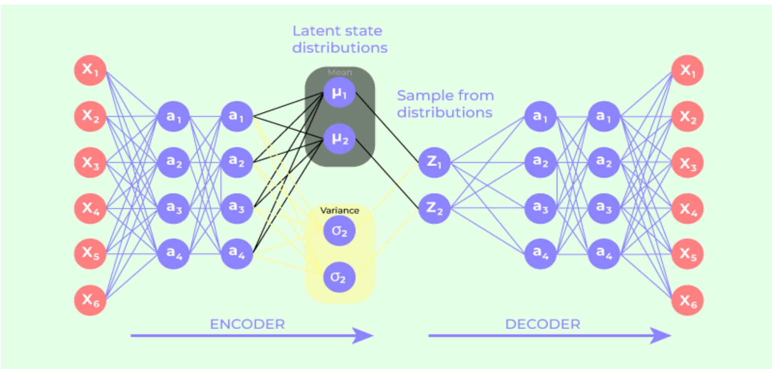


Figure 1.9 Variational autoencoder

1.5.4 Stacked AE

A Stacked Autoencoder is a type of neural network that has many layers of autoencoders. The network can create more detailed and abstract images of the input data by using multiple autoencoders stacked on top of each other.

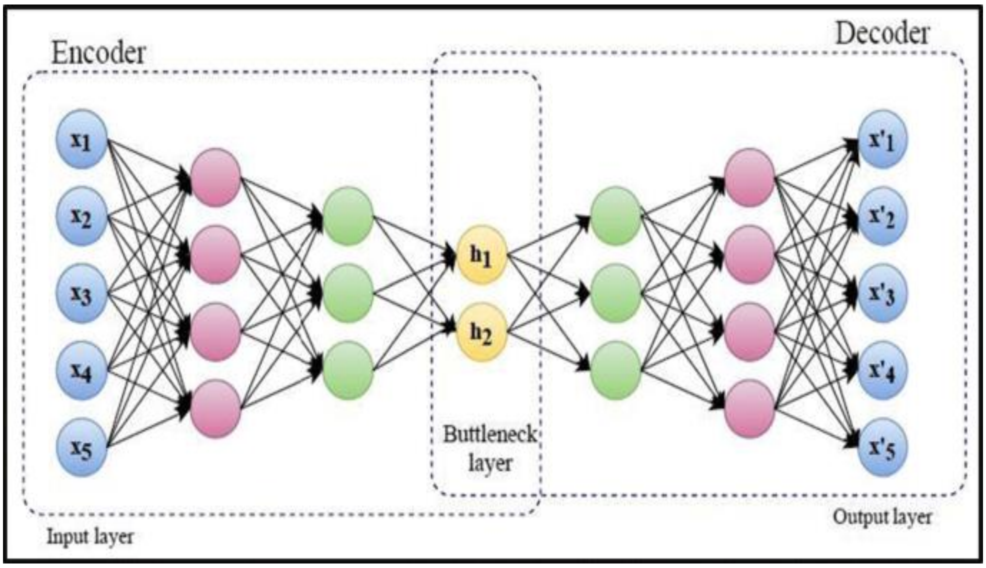


Figure 1.10 Stacked Autoencoder

1.6 The Activation Function

A neuron can be triggered or not. the network will utilize simpler mathematical operations to determine if the input is significant or not. The neural network is similar to the brain in that it has a collection of input signals. The brain uses the type and strength of the signals to decide whether the neuron should be turned on or off. An activation function is a function that decides what a neural network should do with its input. The Activation Function transforms the sum of the node weights into a value that can be used by the next layer of hidden nodes.

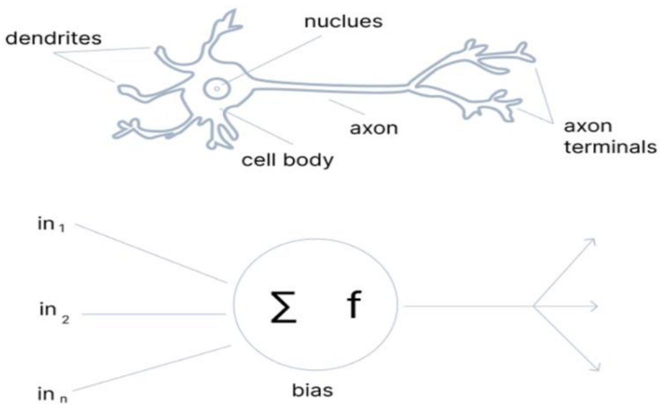


Figure 1.11 Activation function

1.7 Activation Functions in deep NN

1.7.1 Sigmoid

The function takes a real number as an argument and returns a number between 0 and 1. As the input size increases, the probability of the output being 1 also increases. The bigger the input, the less likely the output will be.

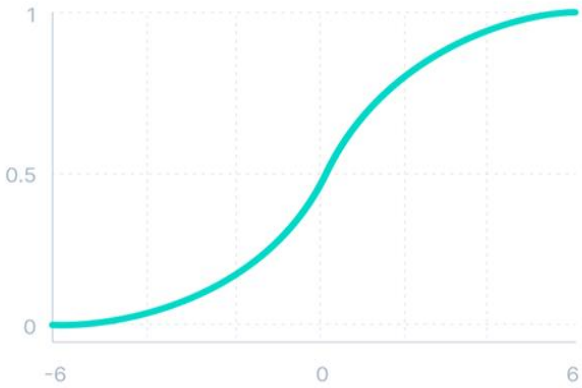


Figure 1.12 Sigmoid

Mathematically, it can be written as:

$$f(x) = \frac{1}{1 + e^{-x}}$$

1.7.2 Softmax

The softmax activation function converts a vector of raw outputs from the neural network into a vector of probability scores.

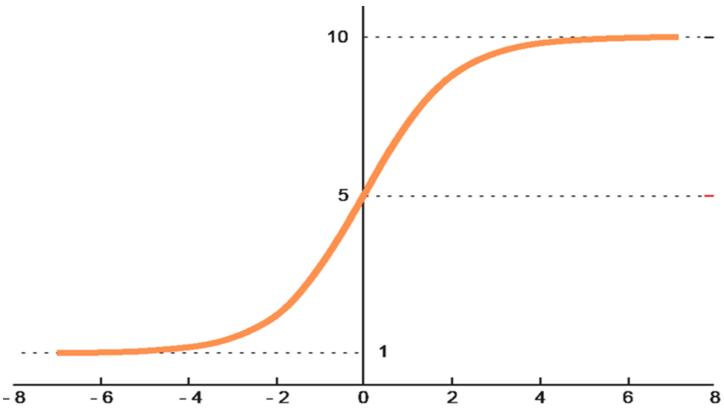


Figure 1.13 Softmax

Mathematically, it can be expressed as:

$$Softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$$

1.7.3 Tanh (Hyperbolic Tangent)

The tanh function is similar to the sigmoid/logistic activation function and has a similar curve, but its output range is limited to -1 to 1. The output value gets closer to 1 when the input value becomes more positive. The output value gets closer and closer to -1 as the input value gets smaller and more negative.

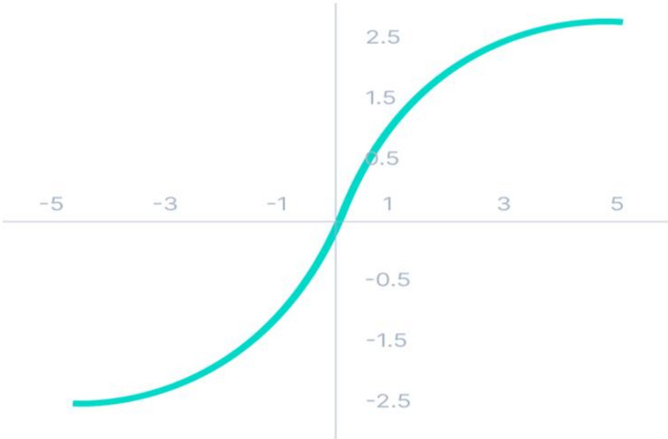


Figure 1.14 Tanh

Mathematically, it can be expressed as:

$$f(y) = \frac{e^{y} - e^{-y}}{e^{y} + e^{-y}}$$

1.7.4 ReLU Function

RLU is an acronym for Rectified Linear Unit. While it may appear as a straightforward linear function, ReLU has a derivative function and facilitates backpropagation, all while being computationally efficient. The crucial aspect to remember is that ReLU does not activate all neurons at once. If the linear transformation makes the neurons output less than zero, they will stop working.

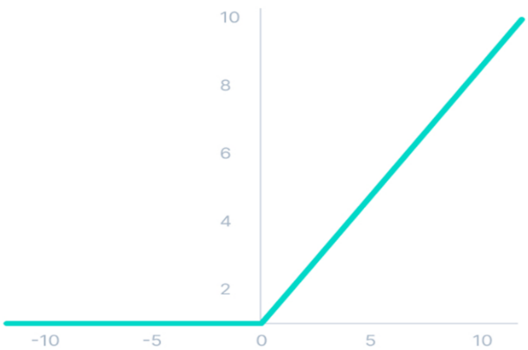


Figure 1.15 ReLU

Mathematically, it can be expressed as:

$$f(x) = \max(0, x)$$

1.7.5 Leaky ReLU Function

The function has a small upward trend in the negative area and is an improved version of the ReLU function. The graph on the left side of the sentence has a non-zero value due to a modification that allows it to handle negative input values, which is the same as the advantages of ReLU.Rephrase We would not come across lifeless neurons in that area.

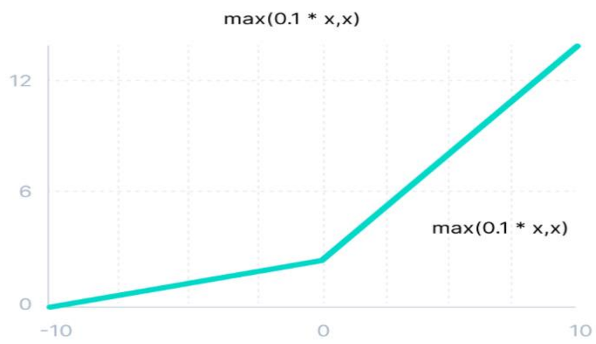


Figure 1.16 Leaky ReLU

Mathematically, it can be expressed as:

$$f(x) = \max(0.1 * x, x)$$

CHAPTER 2

LITERATURE SURVEY

2.1 Overview

As the complexity of users and methodologies in movie recommendation systems has increased, the literature on this topic has also grown more sophisticated. By employing movie recommendation systems, users can have a more personalized experience as these systems suggest films that align with their unique preferences. Traditional methods like collaborative filtering, which predict user preferences by observing similar users or items, have been vital, but they face challenges like data scarcity and the problem of starting with no data. Content-based filters can recommend movies by analysing specific attributes, though it sometimes fails to fully capture user preferences complexity. Hybrid methods that use both collaborative and content-based techniques have shown improved performance in recommending items and handling uncertainty. The popularity of neural collaborative filtering and recurrent neural networks has increased because they can recognize intricate patterns and changes in user behavior over time. Personalization is enhanced by incorporating contextual and social network data in context-aware and social recommendation systems. More precise and personalized movie recommendations are offered by the field.

2.2 Related Work

Collaborative Filtering Recommendation Algorithm Based on User Preference and Optimal Clustering [1] *by Shiyong Xiong, Zebei Wen, Min Zhao.*

Shiyong Xiong et al. [1], suggests an algorithm for collaborative filtering recommendations that takes into account user preferences and optimal grouping. The main areas for improvement are the application of the user-item type preference matrix created by the rating proportion inverse item proportion (RP-IIP) algorithm and the sparrow optimization algorithms optimization of user fuzzy clustering.

Fusion recommendation algorithm based on DeepWalk and Item-Based collaborative filtering [2] *by Yicheng Yu, Rongbin Li, Jun Yin.*

Yicheng Yu et al. [2], suggest an algorithm that addresses the challenges of collaborative filtering by taking into account the temporal aspects of user behaviour, promptly correcting situations where user interest changes over time, and enhancing the algorithm's recommendation accuracy in sparse data scenarios.

A fuzzy recommendation system for predict ing the customers interests using sent iment analysis and ontology in e-commerce [3] *by Karthik, R. V., and Sannasi Ganapathy.*

According to Karthik et al. [3], the evaluation of past customers is a crucial factor in purchase decisions. The new method of using fuzzy logic to analyse online purchases was better than the old one in terms of predicting and simplifying them. The RS used the emotional value of the item that the users purchased to decide its worth.

Sparsity and Mat rix Factorizat ion in Recommender System [4] *by DilipYadav, TusharVaghasiya, SavitaRavate, and Prof. VinitRaut*

According to Dilip Yadav et al. [4], recommendations for related products that are based on computational models are crucial for any type of e-commerce business since they must be able to understand the preferences of the consumer. Availability of purchase determines the goal of the cooperation filtering technique's most recommended item. The author suggests a technique to tackle the issue of data sparsity, which is the lowest rating of items, known as singular value decomposition (SVD).

Design of a Food Recommendat ion System using ADNet algorithm on a Hybrid Data Mining Process [5] *by Wang, Haoxiang*

According to Haoxiang Wang et al. [5], the procedure is more dependable for users due to its precise strategy. As a result, the requirement for developing a trustworthy recommendation system grows daily. Therefore, the neural network-based algorithm is the best choice for solving challenging issues.

Multi criteria collaborative filtering recommender by fusing deep neural network and matrix factorization [6] *by Nassar, Nour, Assef Jafar, and Yasser Rahhal*

The study was conducted by Nassar et al. [6] with the use of two diverse datasets, including movie and travel advisor. When the dataset is based on hotel ratings, the RS used features like rooms, living area, atmosphere, lawn, and so on to affect the ratings. The entire rating for the movie dataset was determined by factors like length, picturization, casting, and so on. Some measures, like the F measure, Mean Absolute Error (MAE), and the proportion of pairs that agree, were not changed.

A personalized context-aware recommender system based on user-item preferences [7] *by Mandheer Singh, Himanshu Sahu, and Neha Sharma.*

According to Singh et al. [7], the conventional methods included both content-based and collaborative filtering strategies. The context-aware recommendation system is an innovative method. Additionally, this strategy separated each other based on the product and client preferences. First, the product was divided according to context values, and then the context values were used to divide the one user into two virtual users.

A Proposed Model to Solve Cold Start Problem using Fuzzy User-Based Clustering [8] *by AL-Bakri, Nadia F., and Sukaina Hassan.*

According to AL-Bakri et al. [8], collaborative filtering is one method that determines how similar products are based on user ratings or reviews when making an online purchase. RS, which stands for Rule-based System, uses fuzzy C to group users according to their own data and then predicts how they will use the products based on their characteristics. The cold start issues were resolved with an explanation of a new model.

Improved System for Performance Evolut ion in Recommendat ion Model [9] *by Sihag, Robin, and Ruchi Rautela*

The Recommendation Model by Rautela et al. [9] noted the cold start problem, data sparsity, and diversity as limitations of current systems and suggested that the hybrid approach is superior to the traditional technique for contextual data and huge user-item correlated data.

Implementation and Evaluation of Movie Recommender Systems Using Collaborative Filtering [10] *by Salam Salloum, Dananjaya Rajamanthri*

Salloum et al. [10] In this study two distinct collaborative filtering techniques are evaluated. Matrix factorization and cosine similarity function-based collaborative filtering with user input are two types of collaborative filtering. We also introduce a new hybrid method in this thesis that defines the cosine similarity function with age, genre, and date included.

Collaborative Filtering with Temporal Features for Movie Recommendation System [11] *by Gopal Behera, Neeta Nain.*

Gopal Behera et al. [11] We proposed a collaborative filtering-based movie recommender system by incorporating temporal features into matrix factorization to handle the dynamic nature of user preference. The model considers the temporal elements and integrates them with the matrix factorization technique to suggest a personalized item to the user.

User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system [12] *by Triyanna Widiyaningtyas, Indriana Hidayah, Teguh B. Adji.*

Widiyaningtyas et al. [12] proposed the UPCSim (User Profile Correlation-based Similarity) algorithm that accommodates user profile data for calculating the similarity weighting to capture the user preference more accurately. In a movie recommendation system, the UPCSim method is used to predict unrated items by classifying comparable user preferences based on k nearest neighbors.

A Comparative Study of Collaborative Movie Recommendation System [13] *by Anjali Ann Joseph and Akhil M. Nair*

Good ratings are required for it to work for a new user who has not rated anything. Since we were able to do a comparative study and deduce the result to develop a recommendation engine[13], our results were found to be accurate to an approximation of 80%, which is a huge benefit to developing an efficient Movie Recommendation System. The primary feature of the Collaborative filtering movie recommendation system is movie rating. It uses a modified cosine similarity matrix to recommend movies. A rating of 1 to 5 is assigned to us. The collaborative filters used to build the recommendation engine have an accuracy of 80%.

Deep Collaborative Filtering Recommendation Algorithm Based on Sentiment Analysis [14] *by Dun Ao and Cong Zhang.*

A new method for processing text data uses sentiment information from the review text. The article extracts the user's feelings about using the item after purchasing it, which helps the user rating to become more reliable[14].A neural network-based collaborative filtering framework was used to model the potential features of users and items. The NCF model was endowed with highly nonlinear characteristics. Comparative experiments showed that using review sentiment information and the SA-NCF model can improve the effectiveness of a system. The sentence can be d as: The article uses the words that express the reviewer feelings as the important data, and it measures how much the reviewers like or dislike the item to find out what the users want and make the recommendation model. According to the experiments, adding sentiment analysis to the user reviews in the recommendation system can improve the models performance and accuracy, and also reduce the training time.

Movie Recommendation System based on User Ratings and Critique [15] *by K. Mahesha, B.T.G.S. Kumara and Kuhaneswaran Banujan.*

This paper presents an innovative approach to enhance content discovery using a sophisticated Content-Based Filtering method[15]. The method analyzes the content of movies using machine learning techniques, focusing on key features such as critique and themes. Professional evaluations, contextual insights, and informed analyses are provided by the critics' reviews. Critics provide a critical lens to the recommendation process, unlike user reviews. This will improve the quality of suggested films for consumers.

We presented a methodology for enhancing movie recommendations[15]. We used rigorous data preparation, LDA topic modeling, and topic assignment to incorporate critics' ideas. The user experience is improved because of the inclusion of critics' perspectives. Our model used advanced techniques to create personalized recommendations for movies. The method was able to balance the evaluations of experts and users. The paper contributes a framework for better movie suggestions.

Recurrent Knowledge Attention Network For Movie Recommendation [16] *by YiHua Cheng, Ying Lu, Na Liu and XiaoJun Tang.*

The current recommendation system is concerned with how to provide personalized recommendation to users and improve the accuracy and user satisfaction[16]. The recommendation system can be enhanced by utilizing the Knowledge Graph. The paper suggests a method for suggesting movies to users based on their knowledge graph and other data. Traditional recommendation methods can be impacted by the lack of user-item interactions and difficulties in making recommendations for new users. The KG is a useful tool for creating recommendation systems that work well with large amounts of data. As a new form of supplementary information, it can help address the issues of having insufficient data and not knowing the initial data.

Movie Recommendation Based on Mood Detection using Deep Learning Approach [17] *by Tahasin Elias, Umma Saima Rahman and Kazi Afrime Ahamed.*

The purpose of this research[17] is to create a system that uses various neural networks to classify human facial expressions such as happy, sad, angry, neutral, disgust, fear, and surprise and then recommend a movie based on that classification. We created our dataset by introducing a couple of faces. We demonstrated a contrast in accuracy between our dataset and the one we gathered to enhance our system. The application for suggesting movies is equipped with the capability to recognize facial expressions. When more complex aggregation is introduced, the investigation of mood detection may be pursued. We have developed a system that incorporates movie recommendation and it has been successful in its implementation. We generated a fresh dataset comprising five categories for identifying human facial expressions, and we also evaluated it against another dataset.

YouTube and Movie Recommendation System Using Machine Learning [18] *by Suraj Suresh K, Mehul Srivastava and Mohana.*

The paper explored various methods to locate relevant documents for scholarly research. Generating and ranking candidates are frequently employed. It is crucial that recommendation systems have the ability to handle real-world problems[18]. Content based and collaborative filtering are frequently employed. Selecting from a huge quantity of web information is a difficulty. User modelling can be made more effective by using mind maps. The proposed model has an accuracy of 98%. The system keeps track of what the user does and what the user chooses, and uses this information to make a user profile. As the user engages with the recommender system, they are also interacting with the webpage, leading to a simultaneous exchange of information. The webpage the user is viewing is regularly updated by the recommendation system. This assistance provides users with more accurate and timely information.

Hybrid Recommendation System with Graph based and Collaborative Filtering Recommendation Systems [19] *by Shubham Khanduri, S. Prabakeran.*

The recommendation systems are Graph based and Collaborative Filtering. A network of the data is created in Graph based[19]. An undirected graph is used to represent the network. The graph is populated with the data and connected to all the movie theater. We will possess a list of movies that have been viewed by users. This could be the users preferred film or the most recent movie they have viewed. For every movie in the list, link prediction is performed. Our objective is to locate the movies that are associated with the movie in order to make predictions about links. A link between two things does not imply a direct relationship, but a shared relationship. By employing classification techniques, we can forecast the rating a user is likely to assign to a movie, even if they have not provided a review. We can obtain a list of highly recommended products by utilizing both of these systems. Our objective is to merge the lists according to the movie’s performance. There will be recommendations after this. The database stores the user's ratings for the movies. The platform stores the user's past activities. This data will be used to suggest movies to the system.

Expanded Autoencoder Recommendation Framework and its Application in Movie Recommendation [20] *by Baolin Yi, Xiaoxuan Shen, Zhaoli Zhang and Jiangbo Shu, Hai Liu.*

The extended autoencoder suggestion framework was proposed in this paper. The commitment to do the proposal is reconstitution with the help of the stacked autoencoders model. The additional details about the situation and the people involved are utilized in the framework to enhance the effectiveness of the suggestions. The suggested structure[20] is connected to the movie suggestion. Initial findings from an accessible database indicate significant differences from conventional approaches. This paper introduces a new method for neural proposal generation using an extended autoencoder structure. The decision to implement the recommendation is to utilize the stacked autoencoders model. To enhance the execution of the proposition, we integrate relevant information about the situation and customers into the framework and apply regularization techniques to the work. The proposed suggestion framework is highly interested in utilizing the extensive side information to expand the autoencoders suggestion structure.

CHAPTER 3

METHODOLOGY

The system we describe is called Collaborative Filtering and is based on Stacked Auto-Encoder. Data preparation, model architecture, training process, and testing are encompassed by the methodology.

3.1 Data preparation

Preparing the data sets is the initial step. We must utilize the library to convert the data into a list. The subsequent step is to convert the lists that were obtained into a list of lists, where each sub-list contains all of the ratings that a single user has provided for all of the films. The preceding list of lists is transformed into a "Tensor" to facilitate the necessary matrix computations. Before machine learning algorithms can give us useful information, datasets usually need some changes to make them fit the algorithms. Some datasets have values that are difficult to change. The information is unusable if it is absent. the programs accuracy will be diminished if the data is incorrect. the necessity of feature enrichment arises from the fact that some datasets are clean but require shaping, while others lack useful business context. the accuracy of model outcomes is improved when data preparation is done in a clean and well-curated manner. Machine learning requires transforming raw data into a usable form, which can be hard because of many problems.

The process of getting data ready for analysis can be difficult due to problems like:.

- **Incomplete or absent documentation:** Obtaining all the necessary data points for a record is a challenging task. The dataset contains null values, unknown values, and a question mark in the missing data:

C	D
age	weight
[50-60)	?
[20-30)	[50-75)
[80-90)	?
[50-60)	?
[50-60)	?
[70-80)	?

Figure 2.1 missing records

- **Unexpected Values or Outliers:** is not uncommon for unusual or unexpected values to appear in a collection of values, particularly when dealing with data that is not well-known or understood.

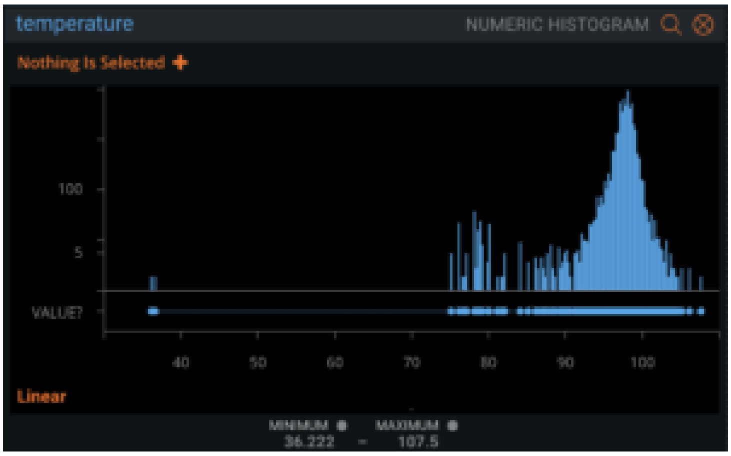


Figure 2.2 Outliers

- **Incorrectly organized / arranged data**: Occasionally the information must be adjusted. One effective approach to tackle this issue is to combine information from various sources.
- **Unreliable values:** Sometimes, the values of the variables in the data we combine may not match. The accuracy of the model can be enhanced by identifying all variations and standardizing them correctly.
- **Few or scarce characteristics:** The number of characteristics or qualities is restricted. We often have to combine data from multiple sources to enhance the features in our data. It can be difficult to merge files from various systems when the data is not properly arranged in columns. Fuzzy matching requires combining various columns to locate the match. It is feasible to merge two datasets that contain customer IDs. As the dataset grows, it becomes more difficult to combine two datasets, one with columns for CUSTOMER first name and CUSTOMER last naand another with a column for CUSTOMER FULL NAME.
- **The requirement for methods like feature engineering:** Techniques like feature engineering require specific features. Even if all the required data is present, the data preparation process may still require techniques like feature engineering to create additional content that will improve the accuracy of the models.

3.2 The dataset employed for development:

3.2.1 Dataset of Users Overview:

Ist Column: user ID

IInd column: Gender

IIIrd Column: Age

1: age < 18

18: age in between 18 to 25.

25: age in between 25 to 34.

35: age in between 35 to 44.

45: age in between 45 to 49.

50: age in between 50 to 55.

56: age > 56.

Fourth Column: Profession

0: not mentioned.

1: faculty or teacher.

2: creator or fine artist.

3: administration

4: students.

5: client service provider.

6: medical profession.

7: manager or superviser.

8: farmer

9: home constructer.

10: school students.

11: legal adviser.

12: Developer.

13: senior citizen.

14: shopkeeper.

15: Scientist.

16: freelancer.

17: architect.

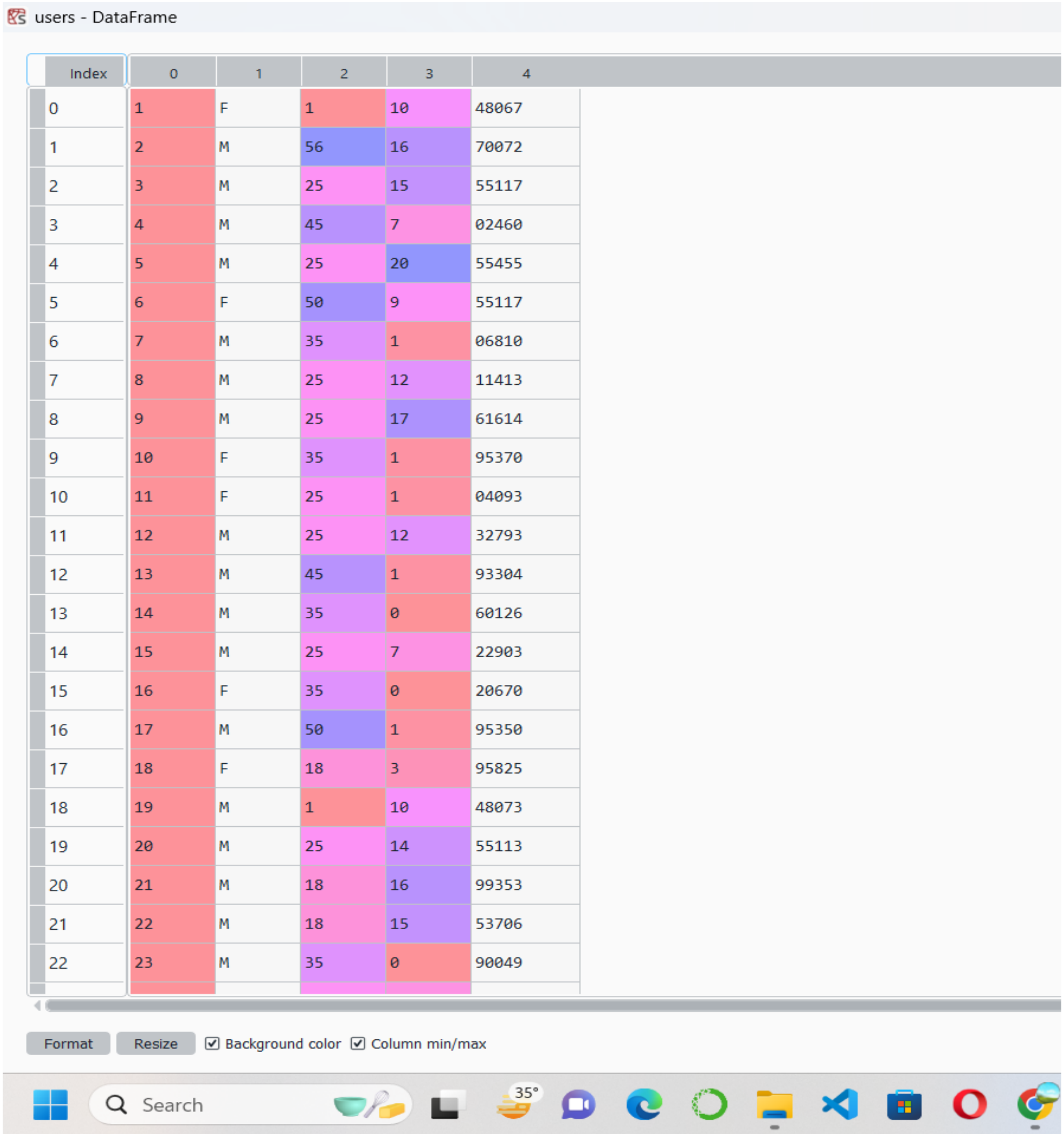
18: craftperson.

19: jobless.

20: blogger.

IVth Column: Zip Code.

Vth Column : Time.



users - DataFrame

Index	0	1	2	3	4
0	1	F	1	10	48067
1	2	M	56	16	70072
2	3	M	25	15	55117
3	4	M	45	7	02460
4	5	M	25	20	55455
5	6	F	50	9	55117
6	7	M	35	1	06810
7	8	M	25	12	11413
8	9	M	25	17	61614
9	10	F	35	1	95370
10	11	F	25	1	04093
11	12	M	25	12	32793
12	13	M	45	1	93304
13	14	M	35	0	60126
14	15	M	25	7	22903
15	16	F	35	0	20670
16	17	M	50	1	95350
17	18	F	18	3	95825
18	19	M	1	10	48073
19	20	M	25	14	55113
20	21	M	18	16	99353
21	22	M	18	15	53706
22	23	M	35	0	90049

Figure 2.3 User Dataset

3.2.2 Dataset of Movies overview:

I^st Column: serial number of movie or id.

II^nd Column: name of movies.

III^rd Column: Genre/Category of movie

Index	0	1	2
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action|Crime|Thriller
6	7	Sabrina (1995)	Comedy|Romance
7	8	Tom and Huck (1995)	Adventure|Children's
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action|Adventure|Thriller
10	11	American President, The (1995)	Comedy|Drama|Romance
11	12	Dracula: Dead and Loving It (1995)	Comedy|Horror
12	13	Balto (1995)	Animation|Children's
13	14	Nixon (1995)	Drama
14	15	Cutthroat Island (1995)	Action|Adventure|Romance
15	16	Casino (1995)	Drama|Thriller
16	17	Sense and Sensibility (1995)	Drama|Romance
17	18	Four Rooms (1995)	Thriller
18	19	Ace Ventura: When Nature Calls (1995)	Comedy
19	20	Money Train (1995)	Action
20	21	Get Shorty (1995)	Action|Comedy|Drama
21	22	Copycat (1995)	Crime|Drama|Thriller
22	23	Assassins (1995)	Thriller

Figure 2.4 Movie Dataset

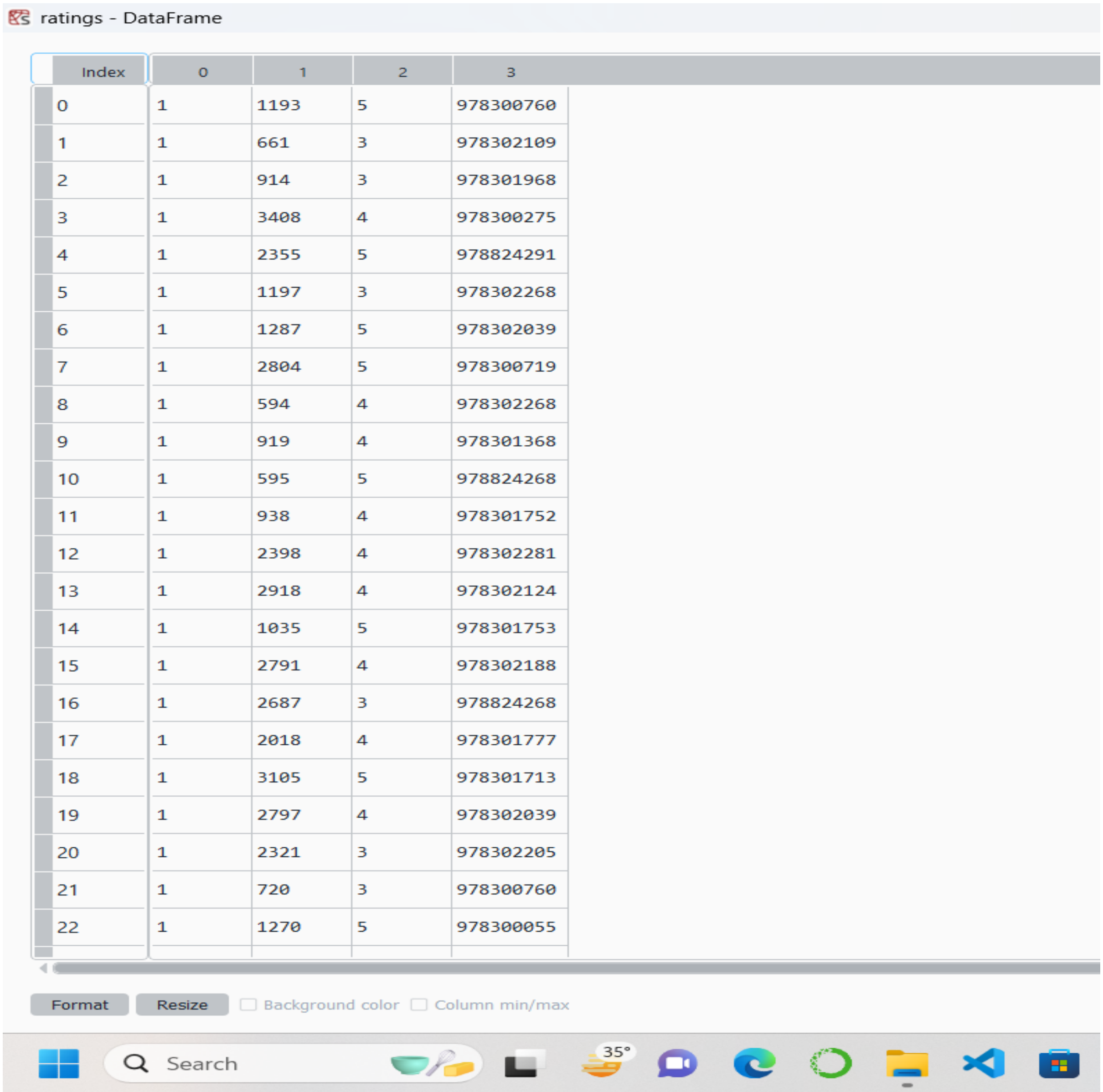
3.2.3 Dataset of Rating given by user

Ist Column: User's ID

IInd Column: Id's of Movies

IIIrd Column: Rating given by user for movies .

IVth Column: Time at which user give rating to a particular movie.



Index	0	1	2	3
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291
5	1	1197	3	978302268
6	1	1287	5	978302039
7	1	2804	5	978300719
8	1	594	4	978302268
9	1	919	4	978301368
10	1	595	5	978824268
11	1	938	4	978301752
12	1	2398	4	978302281
13	1	2918	4	978302124
14	1	1035	5	978301753
15	1	2791	4	978302188
16	1	2687	3	978824268
17	1	2018	4	978301777
18	1	3105	5	978301713
19	1	2797	4	978302039
20	1	2321	3	978302205
21	1	720	3	978300760
22	1	1270	5	978300055

Figure 2.5 Rating Dataset

3.3 System Architecture

The system's overall architecture is constructed in this section. It is a deep auto-encoder, made up of three hidden layers. For this, we establish:

- The visible layer contains a similar number of cells to 1682 movies.
- After performing multiple experiments, it was concluded that the number of neurons in the hidden layer is constant at 20.
- The code or latent layer contains 10 cells.
- Activation function: The Sigmoid and ReLU functions will be tested in this case.
- The learning rate: The learning rate is a crucial aspect of our system design, as it dictates the size of the step we take at each iteration to get closer to the optimal solution. To express the idea that the rate at which a machine learning model acquires knowledge is referred to as its learning speed. The more uncertain the result, the smaller the learning rate. The learning rate was established at 0. 01 in our case.
- The weights (w): The initial value of the connection between the layers of neurons is the weight.

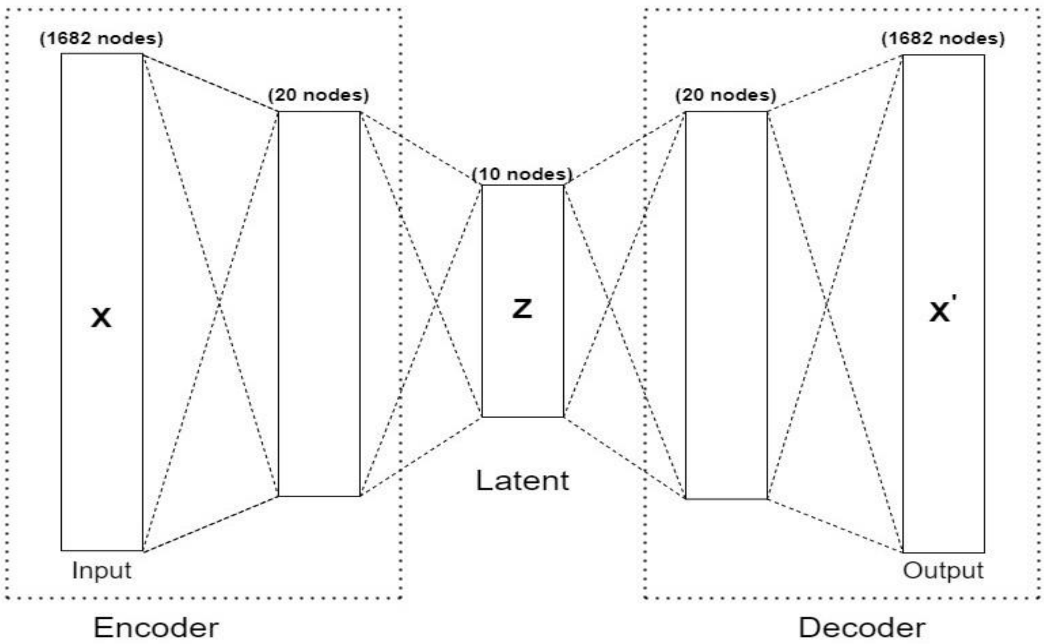


Figure 2.6 The architecture of the system

Prior to training an autoencoder, you must determine 4 hyperparameters.

1. **Code size:** The size of the code is the most significant hyperparameter for adjusting the autoencoder. The amount of data that needs to be compressed depends on how big the bottleneck is. This can be seen as a way to prevent overfitting.

3. **Number of layers in the architecture:** The number of layers in the encoder and decoder of an autoencoder is a significant hyperparameter that requires fine-tuning. Processing information at a shallower level is faster than at a deeper level.

4. **Number of nodes per layer:** The number of nodes in each layer of the network varies with the number of layers. As the number of layers in the autoencoder increases, the input size to each layer decreases, and the number of nodes in each layer also decreases.

5. **Reconstruction Loss:** The effectiveness of the training process for the autoencoder is heavily influenced by the input and output we desire for the autoencoder to adjust to. When working with image data, the two most frequently employed loss functions are Root Mean Squared Error(RMSE) Loss and L1(MAE) Loss. If the values of the inputs and outputs are within the acceptable limits, we can apply the reconstruction loss.

```

class SAE(nn.Module):
    def __init__(self, ):
        super(SAE, self).__init__()
        self.fc1 = nn.Linear(nb_movies, 20)
        self.fc2 = nn.Linear(20, 10)
        self.fc3 = nn.Linear(10, 20)
        self.fc4 = nn.Linear(20, nb_movies)
        self.activation = nn.Sigmoid()
    def forward(self, x):
        x = self.activation(self.fc1(x))
        x = self.activation(self.fc2(x))
        x = self.activation(self.fc3(x))
        x = self.fc4(x)
        return x
sae = SAE()
criterion = nn.MSELoss()
optimizer = optim.RMSprop(sae.parameters(), lr = 0.01, weight_decay = 0.5)
```

Figure 2.7 code snippet of architecture of Autoencoder

3.4 The training

the last step of the models learning process is when it attempts to enhance its ability to provide recommendations. The initial attempt is for the trial and the subsequent one is for examination. the initial two variables that will be utilized in this phase are the count of users who have evaluated a single movie and the loss function. The number of users who have rated at least one movie is determined by the first and second parameters, which are both set to 0. We will execute this within the "epoch" loop.

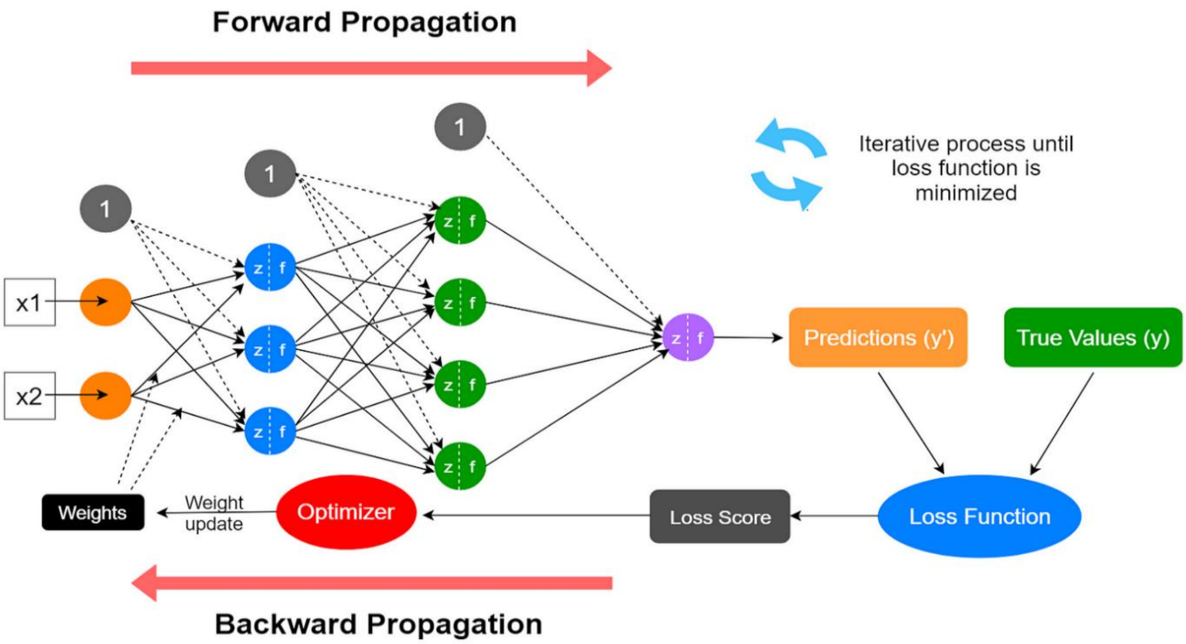


Figure 2.8 training process

The system will be trained in the observation loop. Each user's rating is forecast by the algorithm during the observation iteration. In order to maximize the calculation, the user has to review the movies. The weights are updated when there is a loss. In order to encourage quicker convergence in deep neural networks, we use the RMSprop optimizer at the end of each epoch. Sparse gradients are handled well. We calculate the root mean squared error and mean absolute error for our trained system. The squared deviation of the actual and expected values is computed. The formula for calculating the difference between true and predicted values is squared and it is expressed as shown in equation(3):

$$RMSE = \sqrt{\frac{\sum_{u,i}(V_{u,i} - \hat{V}_{u,i})^2}{N}} \qquad (3)$$

Where N presents the number of ratings used in the test, $\hat{V}_{u,i}$ is the real rating made by user u to item i, and $V_{u,i}$ is the prediction made by the system. When the difference between the predicted and the actual values is small, the model is more correct.

3.4.1 Working of node

The weights in an ANN are numerical values that represent the strength of the connections between the different layers of the network. Every link has a weight that indicates the intensity and orientation of the impact one neuron has on another. The magnitude of the weights applied to the input signal influences the ultimate result of the network.

The output of the network is determined by the weights in an ANN, which have a direct impact on the signals that travel through the network. The ANN modifies its weights over time to predict the correct output for a given set of inputs, during the training phase. The weights in the network demonstrate the knowledge the network has acquired through training.

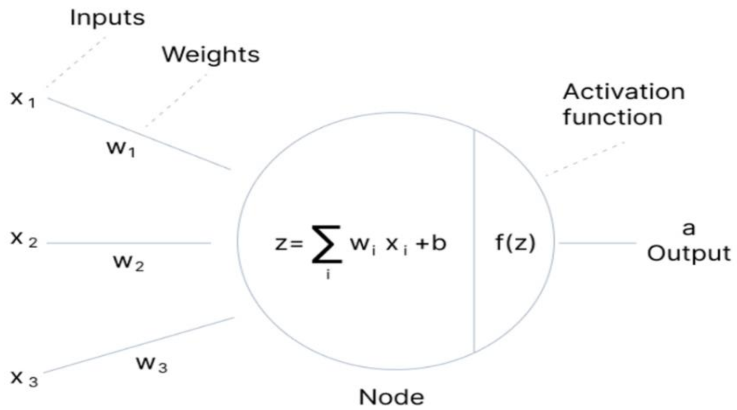


Figure 2.8 working steps of single node

3.4.2 Feed Forward

A feedforward neural network is a basic type of artificial neural network that has no connections that loop in either direction. This network processes data by transferring it from the input to the hidden layers and then to the output layer. The network has no loops or branches. During this stage, the data is inputted into the network. The model incorporates non-linearity by adding up the weighted inputs and applying an activation function to the result. The procedure proceeds until the final layer is attained.

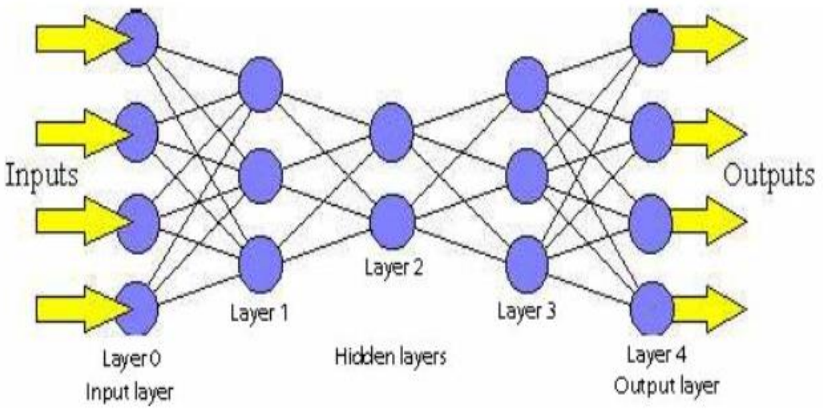


Figure 2.9 Feed forward

Feed-forward networks possess certain qualities.

- The initial and final layers of perceptrons receive inputs and generate outputs. The middle layers are isolated from the external environment.
- Each neuron in the network is linked to every neuron in the subsequent layer. Feed-forward networks are a type of network where data is passed from one layer to the next without any feedback.
- There is no link between perceptrons in the same layer.

3.4.3 Loss Function

The loss function quantifies the discrepancy between the machine learning models predicted outputs and the actual output labels. The model performs better on the training data when the loss function is employed. The machine learning process relies on the loss function to assess the models accuracy and steer the optimization process by highlighting the discrepancies between the models predictions and the actual output labels.

Different ways of measuring how wrong a neural networks predictions are are mean squared error, cross-entropy loss, and hinge loss. The type of model determines the appropriate loss function to use as shown in equation(4).

$$RMSE = \sqrt{\frac{\sum_{u,i}(V_{u,i} - \hat{V}_{u,i})^2}{N}} \qquad (4)$$

The loss function can reveal information about the models performance, such as how well it fits the data or how complex it is. If the training data is not lost, the model will not become too specific to the training data. The model should be able to perform well on unseen data and not lose much of its accuracy on the data it has already seen.

3.4.4 Backpropagation:

Artificial neural networks are trained using backpropagation, which is a method that uses artificial intelligence and machine learning. The computer changes its settings to make the output match the input as closely as possible. Backpropagation is a technique that starts from the output and moves backwards to the input to minimize the error. Static and recurrent are the main forms of backpropagation. In a feed forward neural network, you utilize static backpropagation, where data flows in a unidirectional manner from the input to the output. This method is beneficial when you desire to arrange and categorize. Neural networks can employ static backpropagation.

Neural networks are more active in recurrent backpropagation due to their unique operation. If the data is incorporated into a feedback loop, it can serve as a memory source for future learning. By allowing the network to recognize patterns in the data and make predictions, this enables it to perform sentiment analysis and speech recognition tasks.

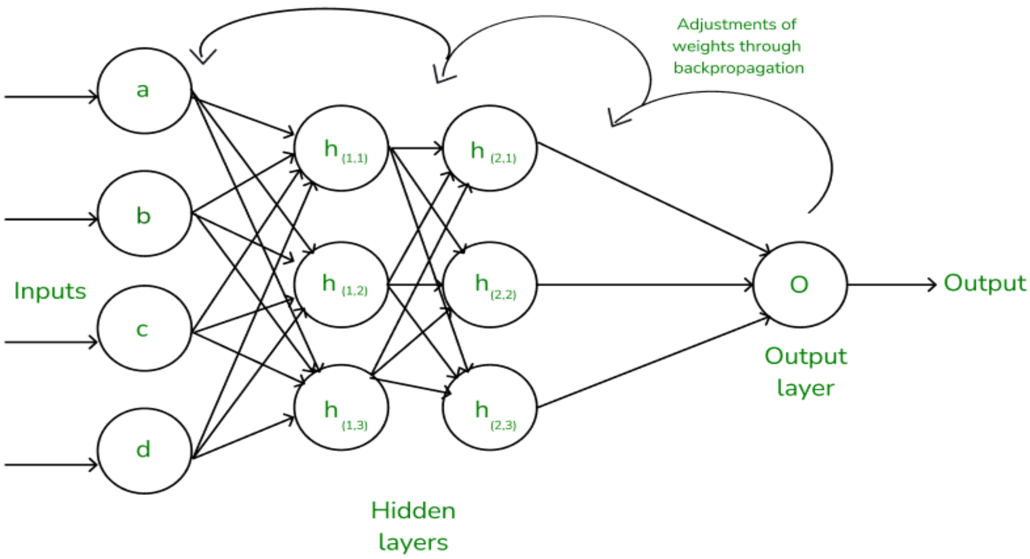


Figure 2.10 Backpropagation

3.4.5 Optimizer:

Weights and learning rate are attributes of your neural network that can be changed with the help of an Optimizer. Some of the issues encountered with theSGD method in training deep neural networks have been addressed by the RMSProp adaptive learning rate maximization. The problem of a worldwide learning rate is tackled by keeping track of a moving average of the squares of gradients for each weight and then dividing the learning rate by this average. For each weight in the model, the learning rate is modified.

The main concept is to enhance the speed of movement in the flat areas of the loss landscape by minimizing the oscillations in directions with steep gradients. The update changes the way it reduces the learning rate. Rather than keeping all previous squared gradients, the system uses an exponential decay that removes old history, enabling it to converge rapidly once it discovers a bowl. The use of RMSProp is essential for training deep neural networks. SGD encounters certain difficulties, but adaptive learning rates assist in overcoming these issues, resulting in quicker convergence and enhanced stability. Even though it has been successful, it is crucial for practitioners to be aware of its limitations and to take into account the unique requirements of their models and data.

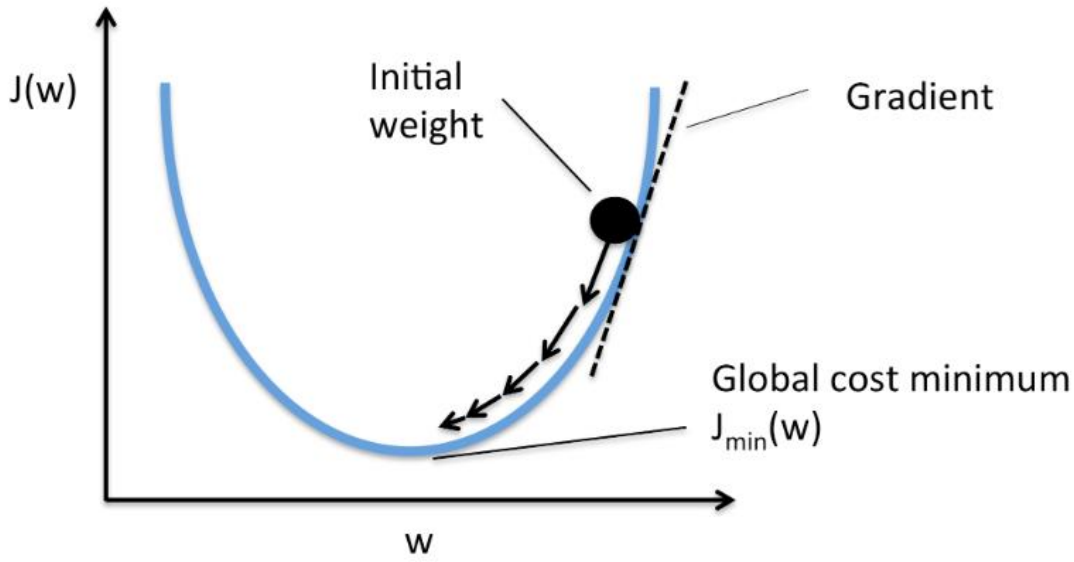


Figure 2.11 Optimizer

3.5 The test

The testing phase is identical to the training phase and we utilize the "test set" data to determine the loss. We only report the loss on the test set for user-rated films. This ensures that our evaluation is not based on previously seen data, leading to a more accurate assessment of the systems performance when faced with new inputs.

By using data that the system hasn't seen during training, the procedure makes sure that its capabilities are evaluated, providing information about its generalization and predictive capabilities There are 3. 6 facilities required.

3.6 Requirements of System:

3.6.1 H/W requirements:

Processor: Intel i5,

RAM: 8GB,

Hard Disk:128 GB

3.6.2 S/W requirements:

Operating System: Windows 10,

IDE: Spyder

Language: Python 3.6,

Libraries Used: Numpy, Pandas, Pytorch ,keras

CHAPTER 4

RESULTS AND ANALYSIS

We tested our idea with some real data to see if it worked. We picked the MovieLens dataset. It is accessible in multiple dimensions. Initially, we employed the ML100K Dataset because it was well-suited for training and its file structure was straightforward. The ML100K folder has many files, such as u. data, which has all the information about the users, 943 of them, who rated 1682 movies. We will show you what we found using the remaining data.

The table shows how well the system did on the test data with the Sigmoid function. The outcomes are determined by the number of iterations. The system evolves, and the epoch increases by 50 each time to keep up with it. The table shows that the Sigmoid function gives the lowest values for the error measures RMSE and MAE. Remember that the smaller the values, the more accurate the prediction.

The same task was repeated with the ReLU function. Table II and the graph on Figure 5 show the results. We noticed that the Sigmoid function gives better results than the ReLU function.

The best recommendation for our system is obtained using the Sigmoid function where the values of the evaluation metrics are RMSE = 0.9678694 and MAE = 0.7555897

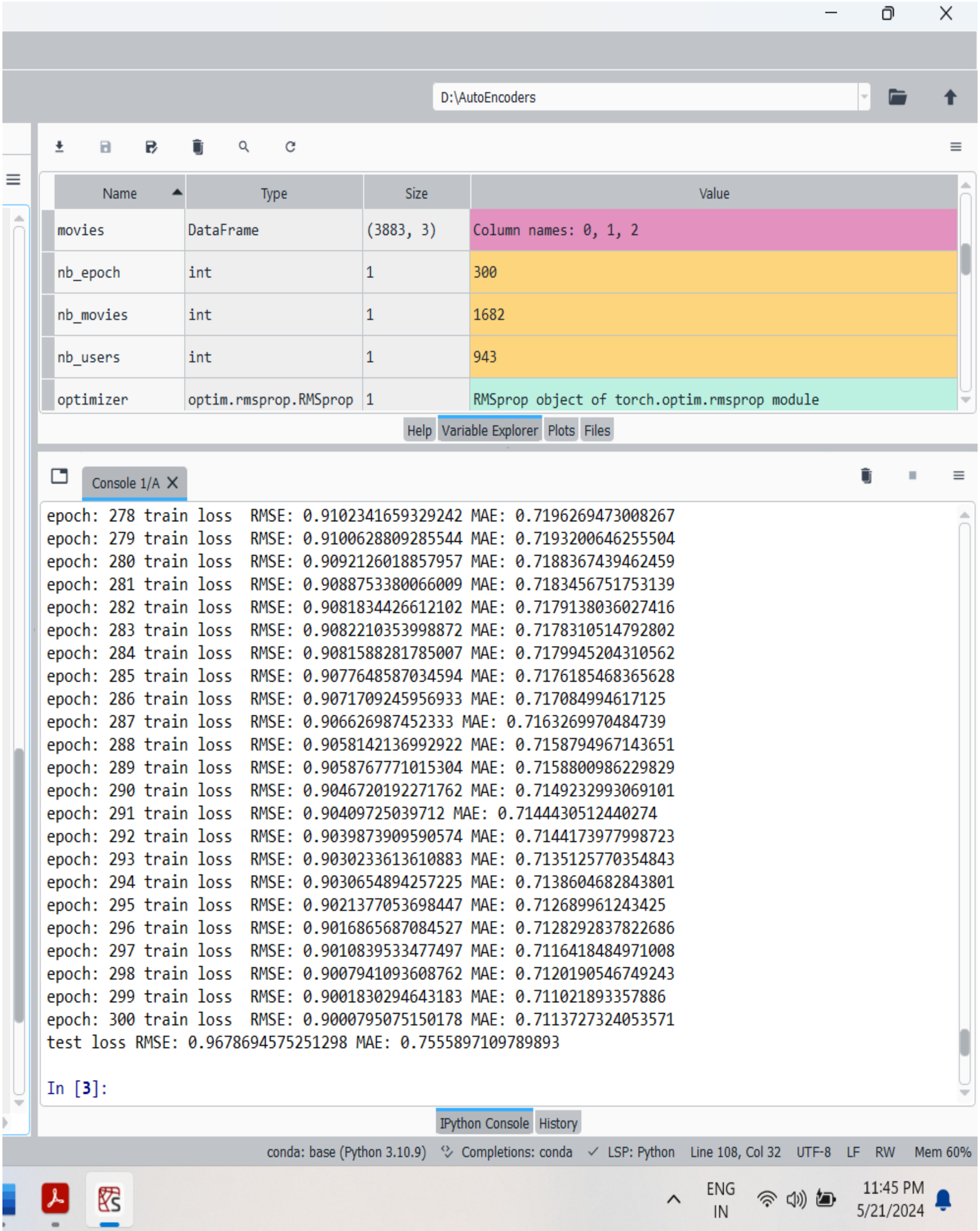


Figure 4.1 Snippet of training loss and test loss

Table 4.1 Result obtained using Sigmoid activation function

Epoch	RMSE	MAE
50	1.0330068	0.8103376
100	0.9929777	0.7873465
150	0.9811114	0.7685639
200	0.9772255	0.7645881
250	0.9922408	0.7811349
300	0.9678694	0.7555897
350	0.9865582	0.7675671
400	0.9819053	0.7651964
450	0.9891206	0.7722704
500	0.9915934	0.7720897

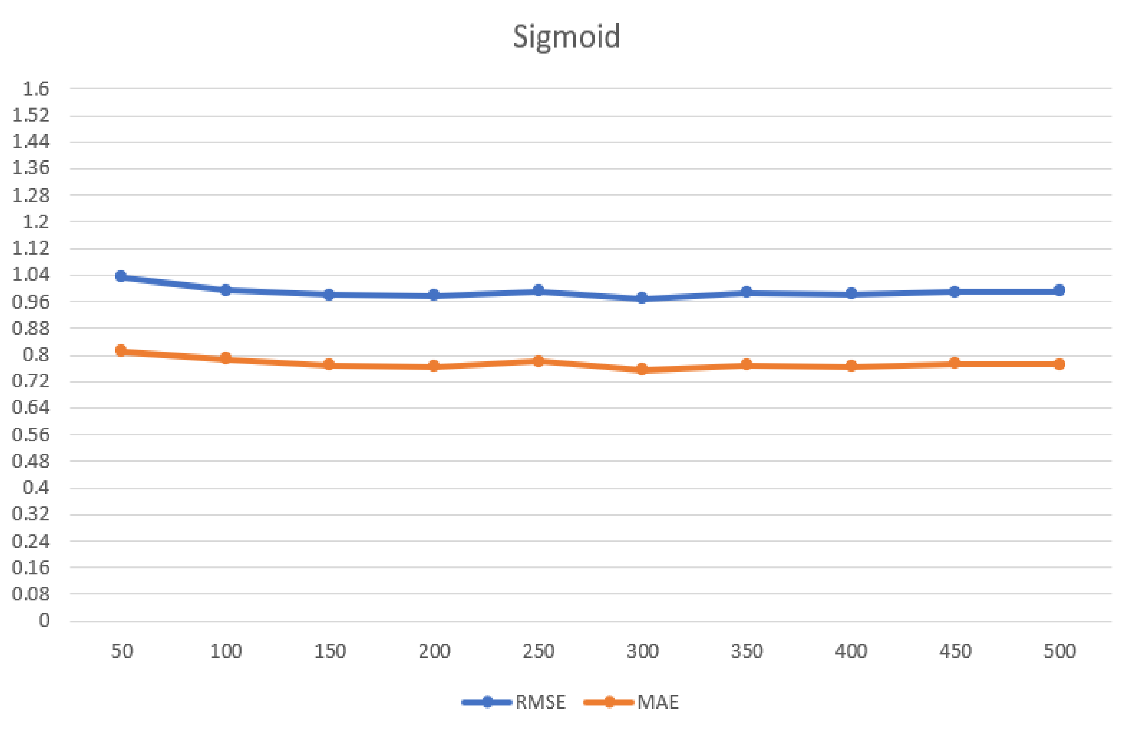


Figure MAE and RMSE for Sigmoid AF

Table 4.2 Result obtained using ReLU activation function

Epoch	RMSE	MAE
50	1.0531605	0.8428935
100	1.0498345	0.8397761
150	1.0509824	0.8438622
200	1.0523657	0.8466483
250	1.0500819	0.8419784
300	1.0492882	0.8418091
350	1.0488006	0.8400243
400	1.0537362	0.8423024
450	1.0944351	0.8659221
500	1.0500763	0.8406346

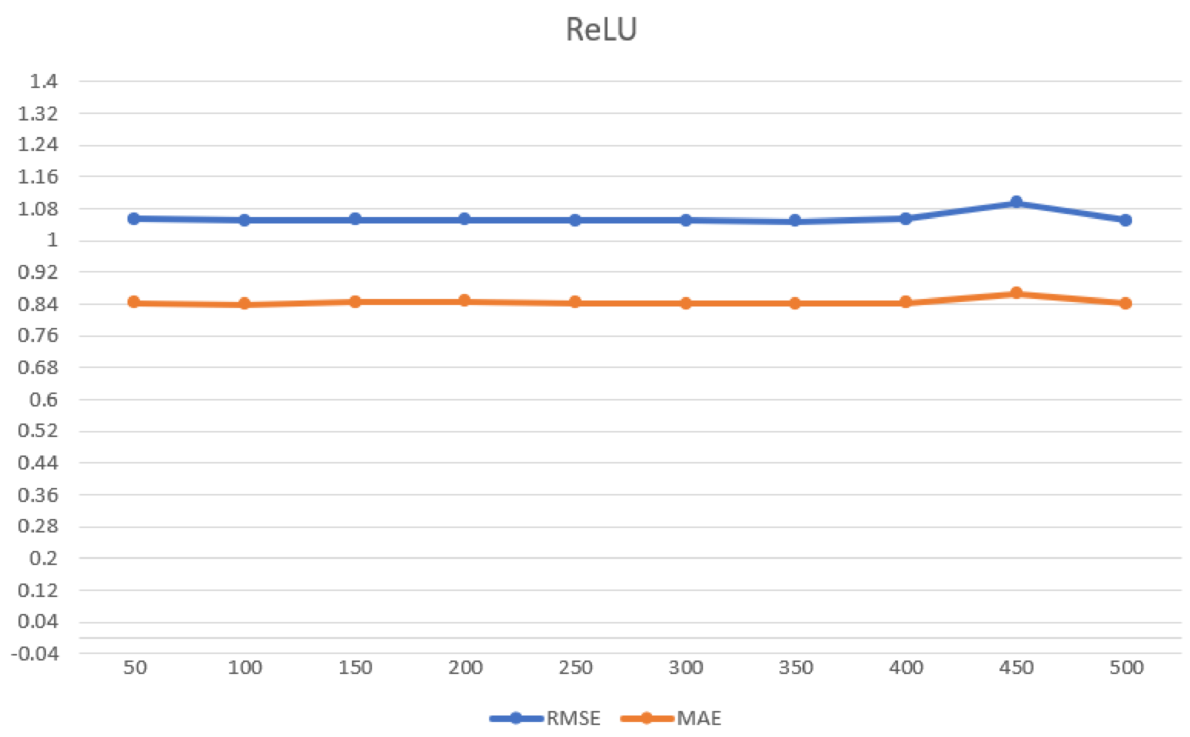


Fig. 4.3. MAE and RMSE in case of ReLU AF

CHAPTER 5

CONCLUSION & FUTURE SCOPE

Autoencoders possess significant potential. They enable you to reveal intricate patterns concealed within your datasets, reduce the number of dimensions in your data, and extract meaningful insights. the performance of recommender systems has been enhanced by deep learning techniques, which have also addressed the challenges they encountered. This paper describes how we applied a sophisticated neural network that can learn to reconstruct its own inputs to improve the performance of our system that suggests movies to users. Our objective has been to determine the function that produces the most favorable results. We employed Rectified Linear Unit (ReLU) and Sigmoid functions in various hidden layers. Sigmoid activation function yields the most favorable outcomes.

The future can be explored by connecting each user's past activities. The model can handle large datasets because each sessions history is added to it. It might be feasible to utilize massive datasets on cloud computing platforms in the future.

REFERENCES

[1] Shiyong Xiong, Zebei Wen, Min Zhao. "Collaborative Filtering Recommendation Algorithm Based on User Preference and Optimal Clustering." In 3rd International Conference on Intelligent Communications and Computing(ICC), 2023.

[2] Yicheng Yu, Rongbin Li, Jun Yin. "Fusion recommendation algorithm based on DeepWalk and Item-Based collaborative filtering." 3rd International Conference on Computer Science and Management Technology(ICCSMT), 2022.

[3] Karthik, R. V., and SannasiGanapathy. "A fuzzy recommendation system for predict ing the customers interests using sent iment analysis and ontology in e-commerce." Applied Soft Comput ing, 108, p. 107396, 2021.

[4] DilipYadav, TusharVaghasiya, SavitaRavate, and Prof. VinitRaut . "Sparsity and Mat rix Factorizat ion in Recommender System." International Journal of advanced Research in Computer and Communicat ion Engineering (IJARCCE), 2021.

[5] Wang, Haoxiang. "Design of a Food Recommendat ion System using ADNet algorithm on a Hybrid Data Mining Process." Journal: Journal of Soft Computing Paradigm December 2022, no.4 (2021): 272-282.

[6] Nassar, Nour, AssefJafar, and Yasser Rahhal. "Multi-criteria collaborat ive filtering recommender by fusing deep neural network and matrix factorization." Journal of Big Data 7.1, pp. 1-12, 2020.

[7] Mandheer Singh, Himanshu Sahu, and Neha Sharma. "A personalized context-aware recommender system based on user-item preferences." In Data Management , Analyt ics and Innovat ion, Springer, Singapore, pp. 357-374, 2019.

[8] AL-Bakri, Nadia F., and Sukaina Hassan. "A Proposed Model to Solve Cold Start Problem using Fuzzy User-Based Clustering." In 2019 2nd Scient ific Conference of Computer Sciences (SCCS), IEEE, pp. 121-125, 2019.

[9] Sihag, Robin, and Ruchi Rautela. "Improved System for Performance Evolut ion in Recommendat ion Model." (2020).

[10] Salam Salloum, Dananjaya Rajamanthri "Implementation and Evaluation of Movie Recommender Systems Using Collaborative Filtering" Journal of Advances in Information Technology Vol. 12, No. 3, August 2021.

[11] Gopal Behera, Neeta Nain,“Collaborative Filtering with Temporal Features for Movie Recommendation System” International conference on Machine Learning and Data Engineering, Science Direct 218(2023) 1366-1373.

[12] Triyanna Widiyaningtyas, Indriana Hidayah, Teguh B. Adji, “User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system”, Journal of Big Data, 2021 – Springer Volume 8, article number 52.

[13] Anjali Ann Joseph and Akhil M. Nair, “A Comparative Study of Collaborative Movie Recommendation System” IEEE, 2022 International Conference on Electronics and Renewable Systems (ICEARS).

[14] Dun Ao and Cong Zhang,IEEE, “Deep Collaborative Filtering Recommendation Algorithm Based on Sentiment Analysis”,2023, 8th International Conference on Information Systems Engineering (ICISE).

[15] K. Mahesha, B.T.G.S. Kumara and Kuhaneswaran Banujan, “Movie Recommendation System based on User Ratings and Critique”, 2024, 4th International Conference on Advanced Research in Computing (ICARC).

[16] YiHua Cheng, Ying Lu, Na Liu and XiaoJun Tang, “Recurrent Knowledge Attention Network For Movie Recommendation”,IEEE, 2020 3rd International Conference on Electron Device and Mechanical Engineering (ICEDME).

[17] Tahasin Elias, Umma Saima Rahman and Kazi Afrime Ahamed, “Movie Recommendation Based on Mood Detection using Deep Learning Approach”,IEEE, 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT).

[18] Suraj Suresh K, Mehul Srivastava and Mohana, “YouTube and Movie Recommendation System Using Machine Learning”,IEEE, 2023 Second International Conference on Electronics and Renewable Systems (ICEARS).

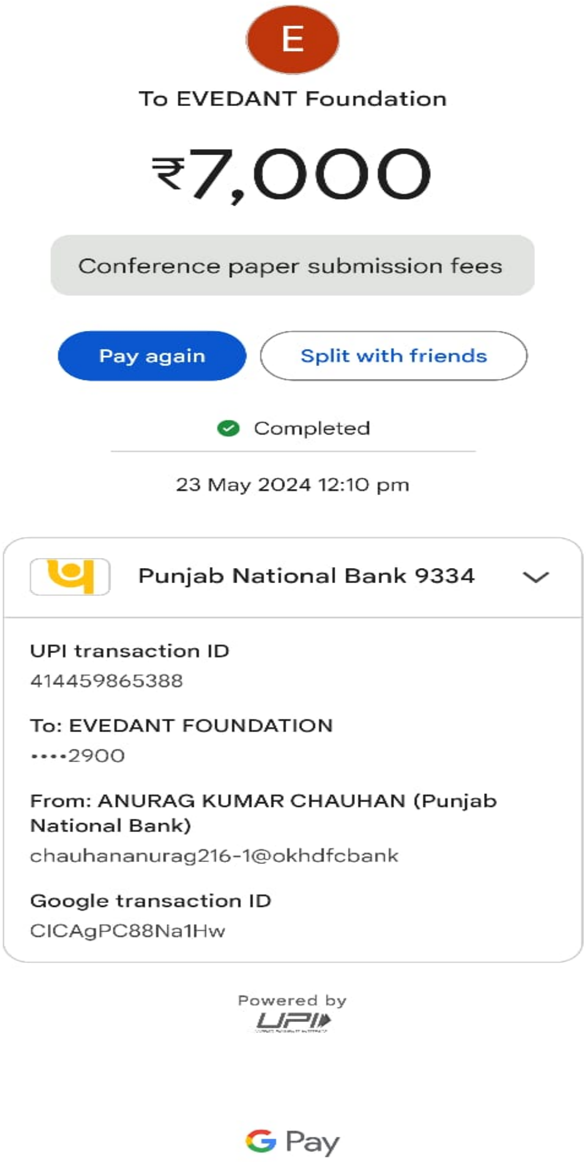
[19] Shubham Khanduri, S. Prabakeran, “Hybrid Recommendation System with Graph based and Collaborative Filtering Recommendation Systems”,IEEE, 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon).

[20] Baolin Yi, Xiaoxuan Shen, Zhaoli Zhang and Jiangbo Shu, Hai Liu, “Expanded Autoencoder Recommendation Framework and its Application in Movie Recommendation”, IEEE, national Engineering Research Center for e-Learning, Central China Normal University, Wuhan Hubei, China.

LIST OF PUBLICATION AND THEIR PROOF

1. Anurag Kumar Chauhan, "A Review on Recommendation Systems Utilizing Diverse Collaborative Filtering Algorithms", Accepted at **"International Conference on Intelligent Computing and Communication Techniques (ICICCT)"**, at JNU New Delhi, India.

   Paper Id: 672



Notification of acceptance of paper id 672
1 message

Microsoft CMT <email@msr-cmt.org> Fri, May 10, 2024 at 1:32 PM
Reply-to: ICICCT Conference <contactus@icicct.in>
To: ANURAG KUMAR CHAUHAN <chauhananurag216@gmail.com>

Dear Dr./ Prof. ANURAG KUMAR CHAUHAN,

Congratulations...

Your paper / article paper id 672: A Review on Recommendation Systems Utilizing Diverse Collaborative Filtering Algorithms has been accepted for publication in International Conference on Intelligent Computing and Communication Techniques at JNU New Delhi, India.

Kindly save your paper by given paper id only (eg. 346.docx, 346.pdf, 346_copyright.pdf)

Registration Link:

https://forms.gle/mSsHa8GMLtMkWuaq8

Please ensure the following before registration and uploading camera ready paper.
1. Paper must be in Taylor and Frances Format.
Template and copyright with author instruction are given in below link: https://icicct.in/author_inst.html

2. Minimum 12 references should be cited in the paper and all references must be cited in the body. Please follow the template.
3. The typographical and grammatical errors must be carefully looked at your end.
4. Complete the copyright form (available at template folder).
5. The regular fee (Available in registration section) will be charged up to 6 pages and after that additional Rs.1000 for Indian authors / 10 USD for foreign authors per additional page will be charged.
6. Reduce the Plagiarism below 10% excluding references and AI Plagiarism 0%.
Each Illustration must include a caption and an alternative text description to assist print impaired readers ('Alt Text').
(Alt Text is mandatory for each Illustrations)
7. Certificate will be issued by the name of registered author (Single author only).
8. Certificates may be issued to all other authors on the extra payment of 1000/- INR per author.
9. Last Date of registration and uploading copyright and camera-ready copy: 31/05/2024.
10. Make a single payment which includes registration fee + Extra certificates fee + Extra page fees.
11. Permissions: Kindly make sure the permissions for each copyrighted artwork file have been cleared ahead of the submission, with the details listed in the Permission Verification form (attached). All permission grants must be submitted along with your final manuscript.
Figures: Please make sure no figures are missing, and all figures are high resolution
Tables: Please ensure that there are no missing tables, and the tables in your manuscript are not pasted as figures
Citation: Kindly ensure there are no missing citations in your manuscript

Registration Link: https://forms.gle/mSsHa8GMLtMkWuaq8

Registration Fee to be deposited in below account

Bank Account Details :

Indian Account Details:

Account Holder Name: EVEDANT Foundation

Account Number: 0674002190422900

IFSC Code: PUNB0067400
SWIFT Code: PUNBINBBGNM

Branch: Punjab National Bank, Navyug Market, Ghaziabad

Account Type: Current Account

International Conference on Intelligent Computing and Communication Techniques
28 & 29 June 2024.

Thanks and Regards
Convener
International Conference on Intelligent Computing and Communication Techniques
contact details : icicctcon@gmail.com, https://icicct.in/index.html

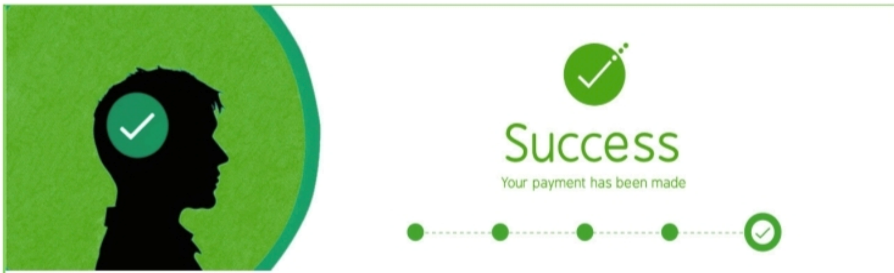
2. Anurag Kumar Chauhan, “Employing Autoencoders in Building a Movie Recommendation System” Accepted at **“National Conference on Advanced Computer Science and Information Technology (NCACSI - 24)”,** Udaipur, India.

Paper Id: National Conference_8175362

Transaction Acknowledgment - Successful Payment

1 message

National Conference <info@nationalconference.in> Wed, May 22, 2024 at 11:54 AM
Reply-to: National Conference <info@nationalconference.in>
To: Anurag Kumar Chauhan <chauhananurag216@gmail.com>



Dear Anurag Kumar Chauhan,

We are pleased to inform you that your online payment transaction has been successfully processed. The details of your payment are as follows:

Order ID: 664d8dd87035b

Reference ID: 113297090156

Amount: 5125.00 INR

Name: Anurag Kumar Chauhan

Email ID: chauhananurag216@gmail.com

Phone Number: 6390048362

Thank you for choosing **National Conference**
. We appreciate your prompt payment.

For assistance or inquiries, feel free to reach out to us at info@nationalconference.in or **+91 9677007228**.

Best Regards,
National Conference
info@nationalconference.in
+91 9677007228

National Conference on Advanced Computer Science and Information Technology (NCACSI - 24) Acceptance & Next Steps
1 message

National Conference <info@nationalconference.in> Wed, May 15, 2024 at 5:31 PM
To: chauhananurag216@gmail.com

Dear **Anurag Chauhan**,

Greetings!

We are delighted to inform you that your research article titled "Employing Autoencoders in Building a Movie Recommendation System" has been accepted for presentation at the upcoming "National Conference on Advanced Computer Science and Information Technology (NCACSI - 24)" scheduled for 15th June 2024 in Udaipur, India.

To progress further, we kindly ask you to move on to the conference registration phase through the provided link:

Conference Registration Link: https://www.nationalconference.in/event/registration.php?id=2482541

For an immediate registration experience, our team is excited to offer you a special coupon code "**INSTANT250**" providing an **INR 250** concession on your registration fee.

Act promptly, as this offer remains valid until the **19th May 2024**. Ensure your registration is completed, and the coupon code is applied before the deadline for maximum savings.

Here's a concise guide on redeeming your coupon code:

Step 1: Click "Register Now," select the relevant fee, and complete the form with your details.

Step 2: Enter the online gateway, furnish the necessary information, confirm, and proceed to the payment checkout.

Step 3: Locate the coupon code section on the CC Avenue checkout page. Enter the coupon code given above and click "Apply" to access the discount.

Step 4: Input your card details and finalize the authentication process.

Payment Confirmation: After completing the registration, share the payment confirmation with us.

For any questions or assistance, feel free to contact us via email. You can also reach us at **+91 9677007228.**

--

Thanks & Regards,

Raj Kumar

Coordinator

National Conference

info@nationalconference.in

www.nationalconference.in

Contact: +919677007228

NATIONAL CONFERENCE

Unsubscribe If you no longer wish to receive mail from us.

PAPER NAME

anurag_2k22cse05.pdf

WORD COUNT 8208 Words	CHARACTER COUNT 43409 Characters
PAGE COUNT 42 Pages	FILE SIZE 1.9MB
SUBMISSION DATE May 23, 2024 11:31 AM GMT+5:30	REPORT DATE May 23, 2024 11:32 AM GMT+5:30

5% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 2% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 3% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Cited material
- Small Matches (Less then 8 words)

anurag_2k22cse05.pdf

My Files

My Files

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:59816182

Submission Date

May 23, 2024, 11:31 AM GMT+5:30

Download Date

May 23, 2024, 11:34 AM GMT+5:30

File Name

anurag_2k22cse05.pdf

File Size

1.9 MB

42 Pages

8,208 Words

43,409 Characters

How much of this submission has been generated by AI?

0%

of qualifying text in this submission has been determined to be generated by AI.

Caution: Percentage may not indicate academic misconduct. Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

What does the percentage mean?

The percentage shown in the AI writing detection indicator and in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was generated by AI.

Our testing has found that there is a higher incidence of false positives when the percentage is less than 20. In order to reduce the likelihood of misinterpretation, the AI indicator will display an asterisk for percentages less than 20 to call attention to the fact that the score is less reliable.

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis **Employing Autoencoders in Building a Movie Recommendation System**

Total Pages **42** Name of the Scholar **Anurag Kumar Chauhan**

Supervisor **Dr Vinod Kumar**

Department **Computer Science and Engineering**

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: **Turnitin** Similarity Index: **5%** Total Word Count: **8208**

Date

Candidate's Signature **Signature of Supervisor**