Personalization in Federated Recommendation System using SVD++ with Explainability

by

Kabbya Kantam Patwary
17301043
Abid Mohammad Jawad
17301062
Md Tahmid Chowdhury Abir
17201029
Yuma Tabassum Khushbu
17301012

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Kabbya

Kabbya Kantam Patwary
17301043

21.01.22

Abid Mohammad Jawad
17301062

Abir

Md Tahmid Chowdhury Abir
17201029

Yuma Tabassum Khushbu

Yuma Tabassum Khushbu
17301012

Approval

The thesis/project titled “Personalization in Federated Recommendation System using SVD++ with Explainability” submitted by

1. Kabbya Kantam Patwary (17301043)
2. Abid Mohammad Jawad (17301062)
3. Md Tahmid Chowdhury Abir (17201029)
4. Yuma Tabassum Khushbu (17301012)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 16, 2022.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Large-scale distributed Artificial Intelligence (AI) systems are getting more widespread as traditional AI applications require centralizing large amounts of data for training models, posing privacy and security risks. For this reason, the idea of Federated Learning (FL) has emerged where instead of sharing data, the edge devices send model parameters over the network to the global model. Though FL ensures privacy preservation, this system lacks personalization due to the heterogeneous data across the client devices. At the same time, the debate continues over the explainability of the FL model like other AI systems. This paper has implemented SVD++ for movie recommendations using the Movielens 10M dataset to increase personalization in the FL system. Later we have also inaugurated explainability to remove the black-box nature of the recommendation system. To our knowledge, implementing SDV++ for personalization in a federated learning setup has not been introduced before. Our trained model has achieved RMSE value of 0.8906. Finally, ensuring the principles of Responsible AI will make the FL recommendation system more fair and reliable.

Keywords: Federated Learning, SVD++, Responsible AI, Explainable AI

Dedication

This paper is dedicated to our beloved parents for whom we exist.

Acknowledgement

Firstly, all praise to the Great Almighty for whom our thesis have been completed without any major interruption.

Secondly, to our advisor Dr. Md. Golam Rabiul Alam sir for his kind support and advice in our work. He helped us whenever we needed help.

And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI	Artificial Intelligence
CFL	Clustered Federated Learning
$FADL$	Federated-Autonomous Deep Learning
$FedAvg$	Federated Averaging
$FedSGD$	Federated stochastic gradient descent
FL	Federated Learning
$flwr$	Flower Framework
$RMSE$	Root Mean square error
$SHAP$	Shapley additive explanations
SVD	Singular Value Decomposition
XAI	Explainable AI

Chapter 1

Introduction

Artificial intelligence combines several technologies to make machines behave like human intelligence. As a result, humans can put more time into analysis and discover different methods and algorithms. AI systems can perform various tasks that can sometimes exceed human performance. 37% of professional sectors have introduced AI in their system within 2020 [10]. The success of these AI technologies depends on the availability of vast amounts of data, also known as big data. However, it is not always easy to collect big data; instead, we get data of small sizes. Hence, the traditional AI algorithms typically train models using data collected from many edge devices such as mobile phones, laptops, and other computers and brought together on a centralized server. Some people may hesitate to have all their raw data stored in a central location, either because miscreants may hack the central location or because their data is too sensitive to be released. At the same time, in the centralized learning system, the question of ownership over data arises. Besides, there is a concern about privacy and requirements for train data locally [21]. As a result, AI applications need alterations from centralized learning systems to large-scale distributed AI systems, so that training processes do not rely on collecting all data to centralized storage.

Federated Learning emerged as the solution to overcome the problems of centralized learning systems. Federated learning is an example of a decentralized AI model where all the end devices train a model under the supervision of a central server and maintain the decentralization of the training data. The raw data of each client is stored in the edge devices locally. Clients do not exchange data among themselves. In the beginning, the model at the server is either initialized randomly or maybe pretrained on some publicly available data. Then, the devices fetch a copy of the global model from the server at their preferable time stamp. Next, the model is trained using the local data available on that device. In this step, devices send their updates back to the server. The server takes all these model updates from the devices and aggregates them to obtain an improved model. Those updates are stored only for a few minutes, and after being aggregated, those updates get erased. Significantly, here no data is being shifted, only shipping the updates of the model parameters take place. This is how a federated learning system preserves the privacy of data. After aggregation of the model, the improved model is sent to the edge devices again. This training cycle keeps going until we get the desired model. Here the end devices keep their local data and use their locally computed data to train the shared model, ensuring data privacy. The data in the device is related to the model's application[18][16]. The FL system can be classified into two types, based on the number of edge devices and data availability. a) Cross-Device FL, an FL setup with millions of edge devices such as smartphones, laptops, and other personal devices with insufficient data b) Cross- Silo FL, where the number of devices is limited, and the devices are stored with vast amounts of data.

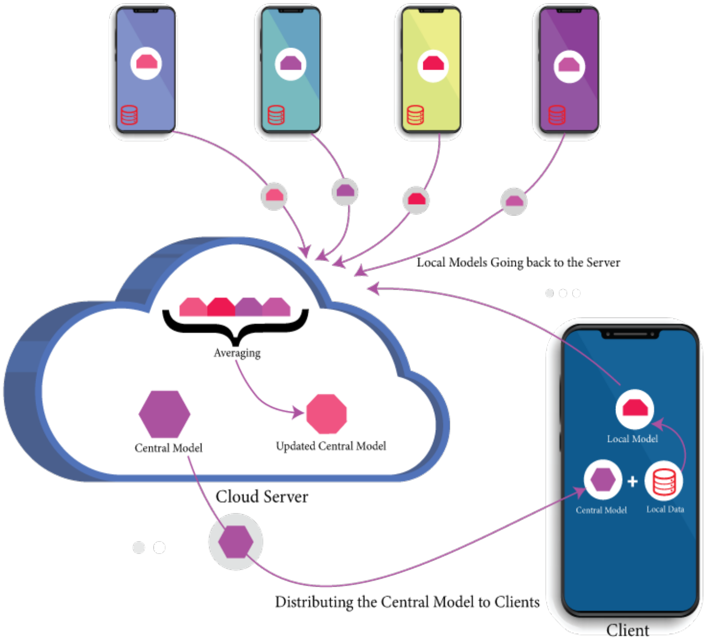


Figure 1.1: Federated Learning System

However, we should not overlook the silver lining amidst so much possibility and prosperity. The question arises: does the AI system act ethically? If an AI system appears to be harmful to society or breaks the laws and regulations, who will be held accused, the one who has built the AI algorithms or the stakeholders? Considering the misuse of AI, the researchers introduced a reliable AI-based system named "Responsible AI." Different companies such as Microsoft, ACM Organization, IEEE set AI principles according to them. The fundamental principles of Responsible AI are more or less the same, such as Explainability, Fairness, Reliability, Data Privacy, Security, Transparency, Accountability. Now the question arises, does FL meet all the principles of Responsible AI? Though the FL system ensures data privacy by not exchanging data between the clients and sending back only the model parameter, the explainability of FL is still questionable, along with the system's fairness.

In the federated learning setup, the clients' data are not similar in their data distribution. [12],[20] This non-i.i.d. data distribution may decrease the performance of the global model. At the same time, this creates fairness issues in the FL setup.
For say, we have a model on next word prediction in the FL setup consisting of 100 clients, where 90 clients are American and the remaining are British. Now if we apply traditional Federated Averaging in this setup, the global model will be heavily biased to the American people due to their large number of participation. As a result, those remaining British users will start getting suggestions based on the

American people's word preference instead of getting their personalization prediction. According to Microsoft, to ensure Responsible AI, they emphasize treating all people equally [22].

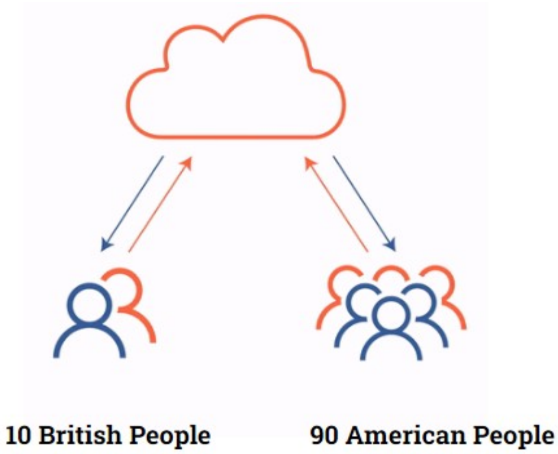


Figure 1.2: Lack of Personalization in FL System

In the context of AI, interpretability means explaining the AI model to humans [2]. In most cases, the process of making decisions by AI systems is unintelligible and mysterious to the average human being. This incomprehensible system is a black box where we train the model and get more accurate output data. To eradicate this uninterpretability, XAI can play a vital role in gaining users' trust [5]. XAI is useful in various domains such as healthcare; where diagnosis and analysis of health data are AI-based, XAI can provide transparency, accountability, and visualize results to validate the model by the physician's knowledge [14], so that model can be improved further. Although simple AI models such as linear regression, logistic regression can be easily understood, they are not as accurate as complex models. Moreover, in complex models, the explanation behind making decisions is still impossible to understand by humans, which makes the AI model uninterpretable. Hence, it opens up an ample opportunity for the researchers to work on the explainability and fairness of the FL system.

1.1 Problem Statement

An AI system should have explainability, reliability, accountability, privacy, and security to meet the principle of Responsible AI [22]. Till now, the federated learning approach is not entirely responsible. The user's data remains in the edge devices rather than being transmitted to the central server in the FL system, ensuring only privacy.

However, due to the heterogeneous data across the client devices, there is a high possibility that a particular group with inadequate data on their local devices will not get the expected suggestion. That is why we have to think about personalization in the FL setup. In the next word prediction, recommendation system personalization is essential to preserve the interest of each group of people. At the same time, it is essential to know how an AI model makes decisions, which can be interpretable by inaugurating Explainable AI.

In our research paper, to ensure fairness for each similar group of people, we have focused on bringing personalization into the movie recommendation system into the FL setup. We have used SVD++ for matrix factorization. Later, we have also included the explainability of our FL setup. To bring explainability, we have implemented Neighbor Style Explainability on the movie recommendation system. This method helps to let us know about the relation between each users. Both the researcher and end-users should know the FL process's working mechanism, such as how the FL process hides sensitive data, ensures fairness for different groups of people, and contributes to decision-making. By including personalization and explainability, we can make the FL model fair and trustworthy.

1.2 Research Objectives

Our paper aims to bring personalization to the FL recommendation setup. In the recommendation system, personalization is essential to have suggestions according to one's preference. We have chosen movielens 10M dataset for the movie recommendation. Due to the non-i.i.d. data distribution, FL may give less preferable suggestions to some groups of people for having insufficient data available in their local devices. Later, we also aim to include XAI in the FL process, which will help us understand personalization better.
Our research objectives are:

- Learn about decentralized learning.
- Understand Federated Learning deeply and its working mechanism.
- Learn about Recommendation Systems.
- Bring personalization in FL setup.
- Discover the black-box nature of the FL system.
- Inclusion of XAI into FL.
- Include the principle of Responsible AI in FL to ensure fairness and interpretability to make the system trustworthy and explainable.

Chapter 2

Literature Review

As soon as data privacy has become one of the biggest concerns, researchers have widely acknowledged federated learning to solve this problem. Nowadays, FL is used in companies like Google, Apple, and Microsoft. However, there are still some issues with this arrangement. One of the most serious issues is the personalization of data. Prior study has been done to prioritize personalization over generalization. Moreover, the explainability of a model is another concern for the researchers. As a result, a number of academics are considering how to introduce explainability in the Federated Learning setup.This section will briefly describe the research works related to our paper.

Sattler et al.[15] presented an improvement of Federated Learning by introducing the "Clustered Federated Learning"(CFL) framework. They described the CFL by a parameter tree where the root node contains a conventional FL model generated by all the clients converging to a stationary point. The next layer splits the clients into two branches or clusters by comparing their cosine similarities. These subgroups again converge into a fixed point. The server saves all the clients' pre-split weight updates so that any new clients can get merged with any leaf cluster that has the highest similarities with them. However, the branching system is terminated when the stationary point fulfills the splitting criteria. They applied CFL to the "permuted labels problem" on the CIFAR-10 dataset for the experiment. There were 20 clients and four permutations. For the first 50 communication rounds, all the clients trained "one model" through traditional FL, and when they reached their goal point, the accuracy was around 20%. After 50 communications, the first split happened, and the accuracy increased by 25%.The splitting occurred again after the 100 and 150 communication rounds so that all the clusters could be separated properly. Also, the separation gap was less than zero in all the clusters, and the test accuracy they acquired was near 60%. As a result, they concluded that the accuracy and performance are much higher by applying the CFL approach after federated learning. Moreover, CFL algorithms work in a privacy-preserving way and can manage population changes over time.

Singhal et al. [23] introduced a model agnostic framework named "Federated Reconstruction" for partially local federated learning. They raised the issue that training a model in a fully global federated way can produce worse performance because of the client's hybrid data distribution. Also, it doesn't maintain users' sensitive data privacy for some settings, such as the matrix factorization model used in the recommendation system. To solve this issue, the authors generated a setting that divided the model parameter into two parts: a global parameter and a local parameter. The local parameters always stay in the client's end devices that ensure data privacy. It also enhances clients' hybrid data performance, reduces communication costs, performs on stateless clients, and provides fast personalization for unseen clients.Each client gets a global variable from the server in all the training rounds and then reconstructs it with their local parameter. After that, they update a reconstructed copy model to the global variable. The server then merges the updates onto the global variable. The experiments on matrix factorization and next-word prediction settings evaluated the framework. They used the collaborative filtering dataset for matrix factorization and ran two different evaluations, StandardEval on seen users and ReconEval on unseen users. The accuracy they got was 43.3%, and FedRecon's performance is better than other approaches. For next word prediction, they used a federated stack overflow dataset. They observed that accuracy proves for smaller size vocabulary.So they concluded that FedREecon provides fast personalization and does well on unseen clients.

Mansour et al.[13] present three approaches for bringing personalization in a federated setup. First, the author proposes a clustering-based system named HypCluster, where the users will be clustered into groups, and the model will be trained for different groups individually. Secondly, the authors suggest a data interpolation method named Dapper, where global data will be combined with local data. Lastly, the model interpolation method named Mapper was introduced where global and local models were mixed together. The authors also claim that each of these methods can be implemented together or individually. Then they implement all three algorithms in the E Mnist Dataset. The HypCluster algorithm gained 88.8% accuracy, whereas the traditional FedAveraging Algorithm gained 84.3% accuracy. Furthermore, HypCluster Algorithm was implemented with Dapper and Mapper, where the accuracy was respectively 90.3% and 90.1%.

Federated Meta-Learning in short FedMeta, a framework for efficient communication, was proposed by Chen et al.[4], where the parameter of algorithms is sent to each client in every iteration instead of the model. When compared to Federated Averaging (FedAvg), FedMeta delivers a reduction in necessary communication cost of 2.82-4.33 times with quicker convergence, as well as an improvement in accuracy of 3.23 percent -14.84 percent. Furthermore, FedMeta protects user privacy by simply transmitting the parameterized algorithm between mobile devices and central computers rather than collecting raw data.

In the paper[7], Arivazhagan et al. propose a model for Personalizing Federated Learning named FedPer. In FedPer, all the edge devices generate two layers of models in each iteration. One is the base layer which is common for all-edge devices like the Federated Averaging Model and shared with all. Another layer is the distinct personalization layer which is kept private in each edge device. Then the authors implemented their proposed model in two different datasets, CIFAR-10 and FLICKR-AES. Using Resnet-34, the accuracy of FedPer was around 86-88%, whereas FedAvg was around 82-84% in CIFAR-10 dataset.

Liu et al.[6], propose a new method, Federated-Autonomous Deep Learning (FADL). This approach uses data from a variety of data sources to train the model, as well as data from specialized data sources. The paper[6] also introduces a balance of global cycles and local epoch. However, they use hospital ICU data, propose FADL over it to distribute local and global training equally and predict the mortality during the ICU admission. They implement a neural network model for prediction purposes. The accuracy of trained the model with FADL is AUCROC of 0.79 and AUCPR of 0.23. The author then compares this result with the traditional centralized learning model and FL model and found that this result is similar to the centralized model but better than the FL model.

Lauritsen et al.[11], propose an explainable AI early warning score (xAI-EWS) system that can detect acute critical illness early. xAI-EWS provides a pictorial explanation of the real-time illness prediction. The system is based on the prediction module TCN and explanation module DTD. From one patient data, it predicts a 78% chance of having Acute kidney injury. Then the DTD explanation module backpropagated towards the input and explained the prediction on the input's value basis. This research work[1] is inspired by the shapley additive explanations (SHAP).

Wang [9] proposes a method for interpreting federated learning with Shapley values, identifying the feature's contribution in vertical FL. It divides the attributes into two parts, one is the guest, and another is the host. They named the interpretation result features important, and using Shapley value, they found that federated guest features have a unified significance value, whereas host features have detailed feature importance. They implement their SHAP Federated algorithm on the Standard Adult Census Income dataset, and their prediction is to check if any individual's income exceeds $50K/yr. They run the SHAP Federated algorithm for the whole feature space, three and five federated guest features. Then generates figures showing the feature importance.

Fiosina [19] proposes horizontal federated learning to process the distributed data while maintaining privacy. The author is implementing this for taxi time travel prediction with explainable Federated learning. The training process isn't dependent on any particular ML algorithm. They also discuss the features of federated learning. Firstly for centralized learning, they found XGBoost as a preferable model that predicted the time-travel with an MSE of 0.00097. Then they use the federated learning approach, which has better performance than XGBoost only if each data source has an equal distribution of data and is run locally by each data provider. The author also wished to implement this approach in the non-identical and non-distributed dataset.

Federated Singular Vector Decomposition, or FedSVD, was proposed by Chai et al.[17] as a masked-based methodology. It conserves the user's private data. The authors compared the standard SVD with the FedSVD. They found that FedSVD generates lossless results, engages low clients, and ensures data confidentiality and Scalability. To evaluate the performance of FedSVD, they used four datasets.They used MNIST, Wine, and Synthetic data to assess lossless legibility.The authors showed that FedSVD's reconstruction loss in raw data was approximately 0.000001% which was low compared with any other privacy-based method. For checking Scalability, they used Synthetic data by considering time consumption. They started with 10k to 50k, created orthogonal matrices, and discovered that the time consumption decreased. They again experiment on 10k to 100k data size, and the time consumption is near to standalone SVD. Moreover, after repeating the test several times,FedSVD maintains the same Scalability. To examine the data protection ability by FedSVD, they used the MNIST dataset and replaced the images with 0's and 1's, which is difficult to interpret. As a result, they concluded that their framework FedSVD is performing well compared to the SVD's existing privacy-preserving methods.

Yehuda Koren [1] merged the Factorization model with Neighborhood in the recommendation system. In the neighborhood method, he focused on the item-oriented approach where the rating is predicted for an item depending on the rating given to the same type of another item by the same user using the Pearson Correlation Coefficient. Then the author used a Matrix Factorization technique named SVD++ using the user-item rating matrix. Then he combined both the neighborhood and matrix factorization models together, allowing both of the models to enrich each other. Then the author implemented the algorithm in the Netflix dataset where the accuracy was 88.70% for 100 factors and 88.68% for 200 factors.

Although all of these studies have contributed in this research area, there still exists some issues with the above mentioned studies. The previous works [4], [7], [13] have raised the similar issue as ours, but their methods can't realistically be used at scale in cross-device environments, because they assumed the client as stateful or always available where they can communicate with the parameters of each clients while training session, which is impractical. Again, in the methods based on clustering [15], there is a high chance for the models to reveal the identity of clients while making clusters, which is a big concern in privacy issues. To solve all these problems, very recently Google introduced Federated Reconstruction[23], a partially local algorithm for federated learning based on matrix factorization. Our study is motivated by their[2] work where we implemented the extended version of Singular Value Decomposition method (SVD++) in a federated setup.

Chapter 3

Proposed Methodology

Our proposed model is designed for recommendation system which will be implemented in federated setup.We will use here SVD++ algorithm, a popular collaborative filtering approach for recommendation system. This algorithm performs better than other algorithms when the dataset is implicit and the level is explicit. To our knowledge, SVD++ have not been implemented yet in Federated Learning setup. Our purpose behind implementing this model is to have the benefits of Federated Learning in Recommendation system, preserving privacy and personalization.

We have used some frameworks to implement our model effectively. First we have used Flower, a popular federated learning framework to setup our environment. Then we used Tf-Rec framework to implement SVD++ in this model. The framework use Keras model to implement the algorithm, where Tensorflow is used to get GPU acceleration which boost the performance on the training process.

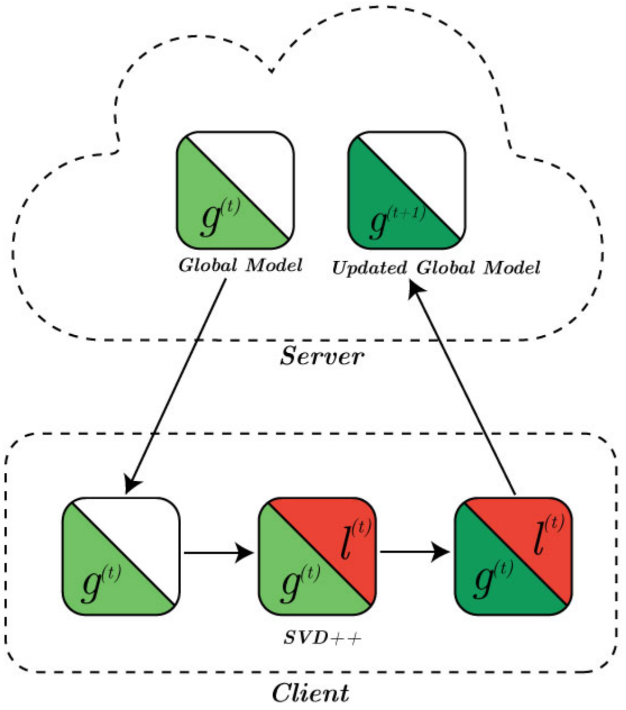


Figure 3.1: Schemetic of our proposed method

Figure 3.1 shows that the server sends the global model $g^{(t)}$ to the clients, then client Factorize the global model with it's local model $l^{(t)}$ using SVD++ and send back the updated global model $g^{(t+1)}$ to the server.

For simulation purpose, we created a local server using the Poetry framework. Here we divide our train data among n number of clients. Clients train these data on their local models and send the weights or parameters to the server. After that, federated averaging is implemented over these local models to make an updated global model in the server. Then the server send the updated global model back to it's clients. Now in client's end, matrix factorization is implemented using SVD++. Here, the global model is factorised with clients local model to make the prediction, which results the client to preserve personalization as well as privacy. Then the global model gets updated and again sent back to the server. This process continues in a iterative way for multiple rounds and gradually improves the global models accuracy.

Chapter 4

Work Plan

This paper aims to successfully implement a personalized movie recommendation system in FL setup, which has been explained using neighbor style explainability. Our model consists of several steps.

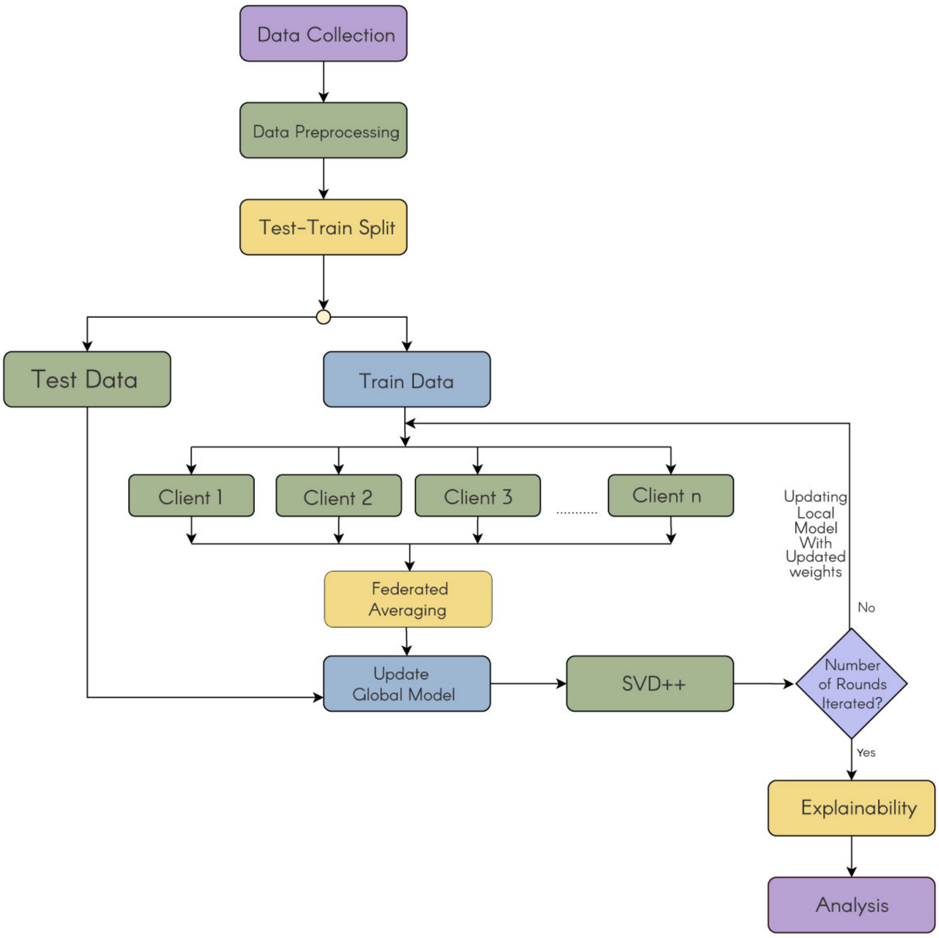


Figure 4.1: Workflow Diagram

At first, we collected a MovieLens dataset, then preprocessed and analyzed the dataset. Then we split the dataset into test and train. We will distribute the training data among all the clients n. The clients will update their local model by using the Federated Averaging algorithm. We update the global model by combining the test data and the model which the clients update through FedAvg. Then we pass the updated model to an improved version of the matrix factorization technique, SVD++. If the iteration number is not complete, then the train data again update the local model with updated weights. When the iteration is complete, we will use neighbor style explainability algorithm to get the explainability of the model and then will analyze the explainability.

Chapter 5

Algorithms and Framework

5.1 Algorithms

We have implemented two prominent algorithms for training and personalizing the recommendation system using the MovieLens dataset. Afterward, we have used the Neighbor Style Explainability algorithm to provide the explainability of the model. In this section, we will briefly discuss the algorithms we have used in our research work.

5.1.1 Federated Averaging

`Federated Averaging algorithm (FedAvg)` is a variant of federated stochastic gradient descent(FedSGD). It is used for averaging the local models to make a better central model and shares its data's weight update, not the gradient update. The algorithm is executed for several rounds of training. At first, the central server generates an initial model and sends the model to a random client set for each training round. Then each client from the client set runs stochastic gradient descent on their local data for E epochs. The updated weights of the client can be expressed as equation 5.1

$$w \leftarrow w - \eta \bigtriangledown \ell(w; b) \tag{5.1}$$

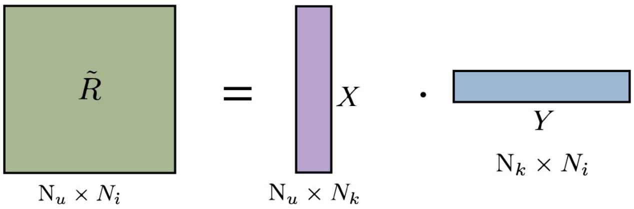
Here, w is the shared model's weight,η is the learning rate, and $\nabla\ell$ is the gradient descent.The weight value will update by subtracting the multiplication of the learning rate and an average of all client's gradients from the current weight.
Then the client sends the updated weights back to the central server. However, the server aggregates all the client's weight and updates its shared model's weight. We can update the weight by using equation 5.2

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k} \tag{5.2}$$

Where,k is each client from the total client K. The client k and shared model's weight update is denoted by w_{t+1}^{k} and w_{t+1}, respectively.The total number of data samples is denoted by n and data sample client k is denoted by n_k.
The shared model's weight will be updated by taking the weighted average, and the work continues in recursion.

5.1.2 Matrix Factorization

Collaborative filtering makes suggestions based on similarities between users and items at the same time depending on their past behaviors. Neighborhood approaches and latent component models are the two main areas of collaborative filtering. The latent factor model is based on Matrix Factorization which works on dimensionality reduction. In this part, we will discuss Singular Value Decomposition(SVD) and the improved version of SVD named SVD++, these are simple and popular machine learning techniques for matrix factorization.
In SVD, the matrix structure is formed where rows are represented as users and columns are represented as items, and ratings given by the users are the elements of the matrix. Using this matrix we can get ratings for every combination of users and items. The factorization of this matrix is done to break it down into small matrices.



$$\tilde{R}_{ui} = \sum_k X_{uk}.Y_{ki}$$

Figure 5.1: Matrix Factorization using SVD

From Figure 5.1, we can see how SVD has factorized the matrices. Here $\tilde{R}$ represents how much rating user u will give to item i. The X matrix is the factor matrix of user u and the Y matrix is the factor matrix for item i. The dot product of these two matrices is the Rating predicted by the user. Here the factor k=1.

SVD++ is the extended version of SVD where along with the matrices, some global effects are also handled precisely.

$$\tilde{R}_{ui} = \mu + bu + bi + \sum_k X_{uk}.Y_{ki} \tag{5.3}$$

Here,μ, bu and bi are the global factors where μ represents the overall average rating, bu represents the user bias and bi represents the item bias.

5.1.3 Explainable AI

Machine learning algorithms are vastly being implemented in many applications nowadays. Critical and complex AI systems are taking over to predict more accurately. Though, in most cases, their process of making decisions is unintelligible and mysterious to the average human being. This incomprehensible system is nothing but a black box where we input data and get output data with higher accuracy. In complex models, the explanation behind making decisions is still impossible to understand by humans. This makes the AI model uninterpretable. In the context of AI, authors [3]define interpretability as the “ability to explain or to present in understandable terms to a human.” To eradicate this un-interpretability, XAI (eXplainable AI) plays a vital role. According to [8], XAI can gain users’ trust, which also summarizes the AI behavior and produces the perception of the black box.

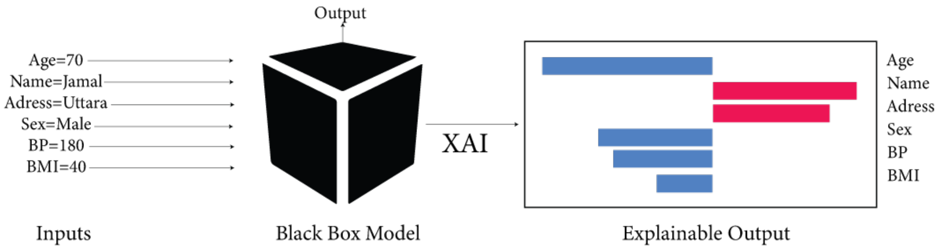


Figure 5.2: Explainability of Black Box Model

The Figure 5.2 shows how we can explain the contribution of each feature for making any prediction. Here the black box can be any Machine Learning Model which takes data as input and predicts an output based on its model, but it is tough for us to know the reason behind that prediction, so here comes XAI which helps to explain the contribution of each feature for making that prediction.

5.1.4 Neighbor Style Explainability

After training the global model, we will have a Rating Matrix (R) where $R_{i,j}$ indicates the prediction of rating, user i will give to movie j. For user based Neighbor Style Explanation, for a particular user i, we will calculate the summation of absolute difference between the rating given by user i and other users for every movie. More formally,

$$\delta_{(i,j)} = \sum_{k=1}^{M} \left| R_{i_k} - R_{j_k} \right| \tag{5.4}$$

For a particular user i, because of which user j, we get the smallest value of $\delta_{(i,j)}$, that j will be the most similar user in respect of user i. This means the prediction made by user i, will be dominated mainly by user j.

5.2 Frameworks

A framework is a template or platform where users can develop their software-based projects. It has a package that includes shared libraries, picture files, reference papers. Users can reform the package whenever needed on a basis of their project. We have used some frameworks to implement our model effectively. In this section we will give a brief description about the frameworks we have used.

5.2.1 Flower

Flower (flwr) is an open-source federated learning framework. It created federated learning in such a way that users can focus on their ML problems without wasting much time. Also, it can handle many clients in real-world problems [24] . According to the flower design principle, the framework does the federation of any deep learning framework by locating at the top of it. So the flower framework is suitable for any Machine Learning(ML) framework such as Keras, PyTorch, and many more. There are four algorithms; FedAvg, Async FedAvg, Fed Prox, Q-Fair FedAvg implemented in the flower framework. Any user can implement their own algorithms whenever needed.

5.2.2 Surprise

Surprise, Simple Python Recommendation System Engine is an open-source Python scikit library. It's used to build and assess any recommendation system that deals with explicit rating data. It has two built-in datasets; MovieLens, Jester, or users can use their dataset. It also includes some prediction algorithms, for instance, matrix explanation, neighborhood method, etc. Here, the inputs are the user and some of the items they rated, including the rating value. The algorithm can interpret the rating of other items for which the users don't provide any rating yet.

5.2.3 Poetry

Some languages have command-line utilities for managing their project. They fulfill the project's requirements by keeping those up to date and managing the installation,updatation, compilation, execution, and execution environment. Python doesn't have such a utility command officially for its project. But a third party made up a utility for managing python projects is "Poetry." Poetry is a system of managing dependency in python projects. Python 2.7 or 3.5+ is needed for poetry and can be used smoothly in any operating system like Windows or Linux.

5.2.4 Tf-Rec

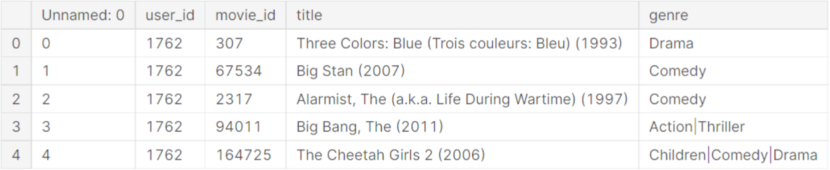
Tf-Rec is an open-source library for generating recommendation systems algorithms. It supports SVD and SVD++ algorithms. Existing frameworks, TensorFlow 2 and Keras can accelerate the training time when the training data size is large. Tf-rec generates an algorithm for implementation, and this can be used by combining it with Tensorflow Code.

Chapter 6

Data Preprocessing and Analysis

6.1 Data Collection

We have collected the “MovieLens 10M” dataset from the Movie Lens site for our research purpose. The dataset contains 10M movie ratings from 71567 users, and the total number of movies is 10681. Users rated the movies from 1 to 5 based on their taste. Each user had to rate at least 20 movies and fill up their demographic information such as age, gender, etc.



	Unnamed: 0	user_id	movie_id	title	genre
0	0	1762	307	Three Colors: Blue (Trois couleurs: Bleu) (1993)	Drama
1	1	1762	67534	Big Stan (2007)	Comedy
2	2	1762	2317	Alarmist, The (a.k.a. Life During Wartime) (1997)	Comedy
3	3	1762	94011	Big Bang, The (2011)	Action|Thriller
4	4	1762	164725	The Cheetah Girls 2 (2006)	Children|Comedy|Drama

Figure 6.1: Dataset

Users who submitted ratings to fewer than 20 films and did not give demographic details have been deleted from the dataset. The dataset contains 10000054 instances. Our dataset contains three data files, “Ratings”, “Movies”, “Tags”. The features of these data files are “User Id”, “Movie Id”, “Movie Ratings”, “Time-Stamp”, “Movie names”, “Released date”, “Movie genre”, and “Key tag”.

6.2 Data Preprocessing

We have connected “Ratings”, “Movies”, “Tags” data files by dictionary-mapping using “movieId” and “userId” features. We have dropped the timestamp from the ratings data file. Then we used sklearn preprocessing for label encoding the features. Again, we used another module from sklearn to convert the features into a min-max scaler where the range was -1 to 1. We shuffled the data files before splitting. We split the dataset into 8:2 ratios for training and testing.

6.3 Data Visualization

By visualizing data, we can examine the data correctly, see the data pattern, and identify any error in data. Also, help during decision-making.
We generate a graph for a visual idea about how many numbers a rating gets. We plot ratings from 0 to 5 on the y-axis and the total number of each rating on the x-axis.

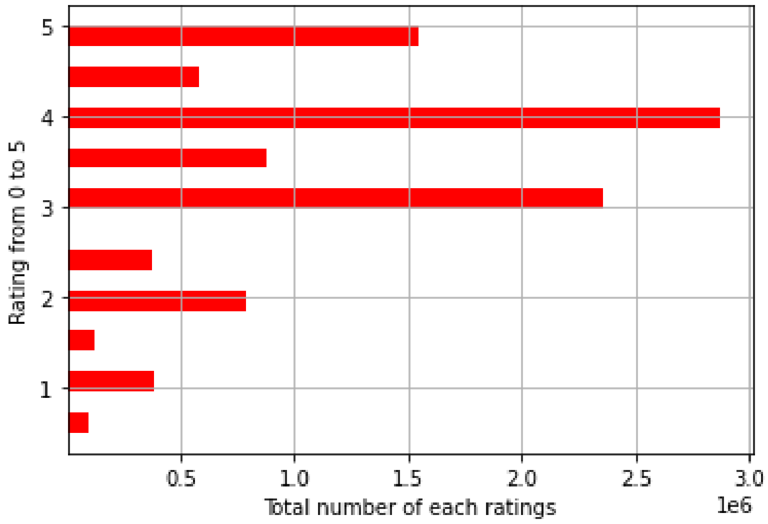


Figure 6.2: Ratings

From the Figure 6.2, we can get the idea that rating 4 gets more than 2.5× 10^6 number, then rating 3 gets more than 2.3×10^6 number.

Then, we generate a graph for understanding the user's view by monitoring the ratings they usually give to the movies.

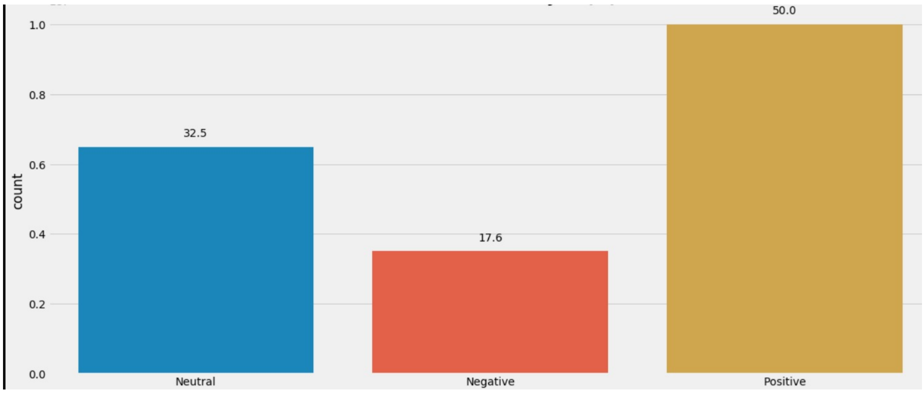


Figure 6.3: User Rating Analysis

Figure 6.3 shows that 50% of users provide a positive rating of 4 or 5 for rating the movies.17.6% users gave e negative rating, which is less than rating 3, and the rest gave a neutral rating of 3.

The following Figure 6.4 presents the movie genre vs. the number of movies released from that genre.

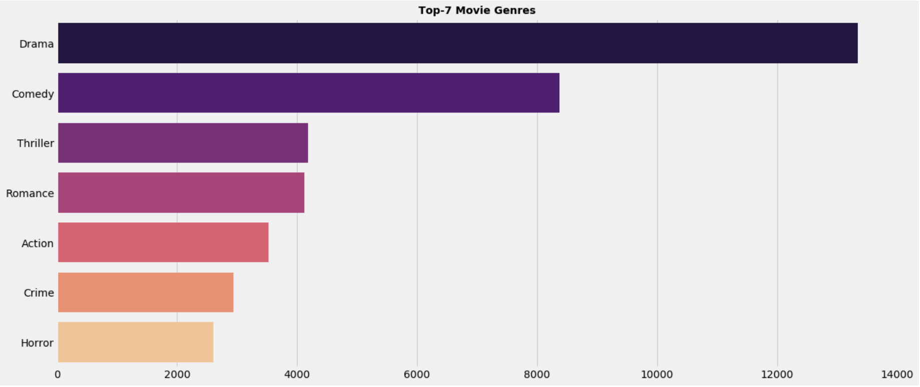


Figure 6.4: Movie genre vs. the number of movies released from that genre

From the Figure 6.4 , we can see more than 12000 movies released from the "Drama" genre and less than 4000 movies released from the "Action" genre.

Our dataset contains movie-related information, so in the graph Figure 6.5, we plot the "movie count" on the y axis and the "movie genre" on the x-axis. It will give us an insight into each genre's movie count.

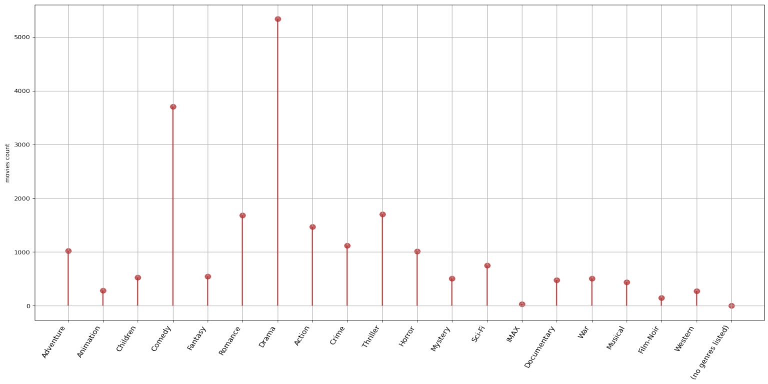


Figure 6.5: Graphical representation of number of movies in each Genres

The Figure 6.5 shows that the data contains more than 5000 "Drama" genre movies and almost 4000 "comedy" movies.

Then, we create a rating vs. year graph. From this graph Figure 6.6, we will know the average rating of each genre for a particular year.

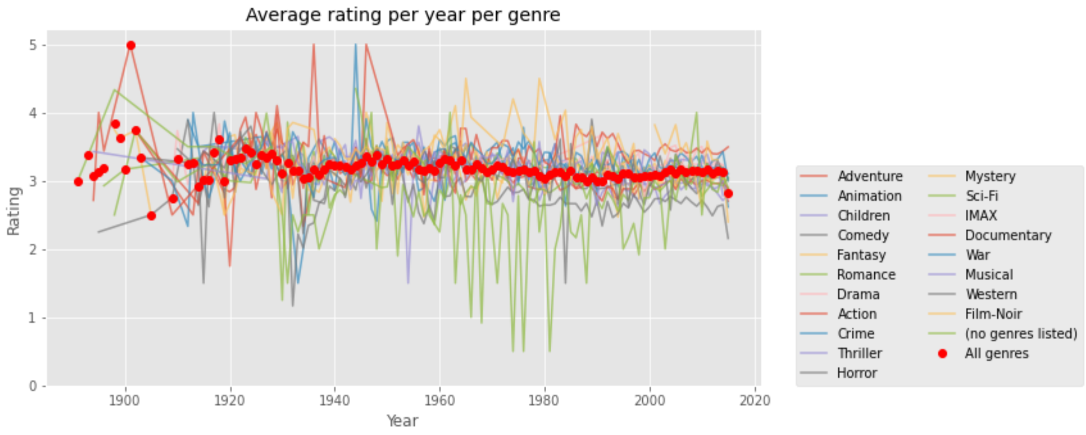


Figure 6.6: Rating vs. Year

Figure 6.6 shows that all the romance genre movies got around 0.5 ratings in 1980. The "red" dot denotes the average rating of all movie genres for any year. For example, let's say in 2017, all the movie genre's average rating was almost 3.

The next graph Figure 6.7 represents the average movie rating with respect to Genres.

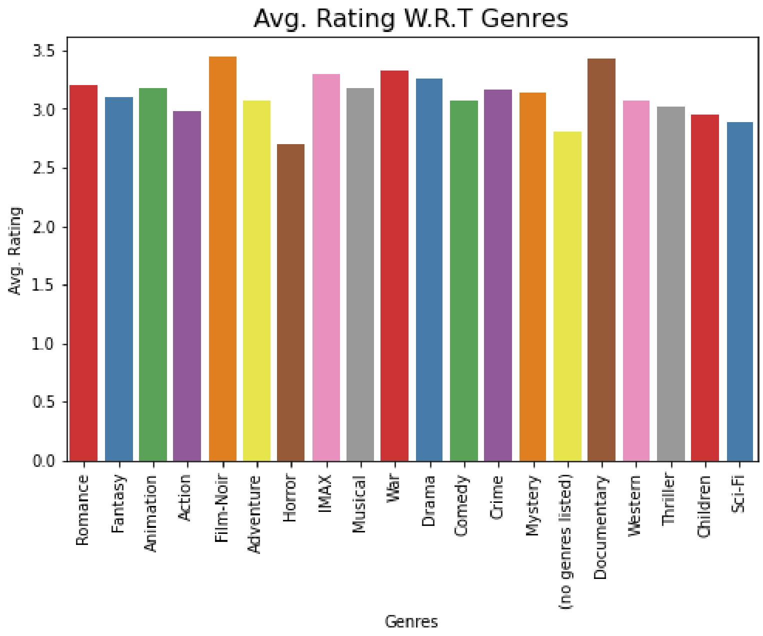


Figure 6.7: Average rating vs. Genres

Figure 6.7 shows that the average movie rating of the Romance genre is around 3.2, and Sci-Fi movies have an almost rating of 3.

Then, we generate a graph for visualizing the number of movies released in a year.

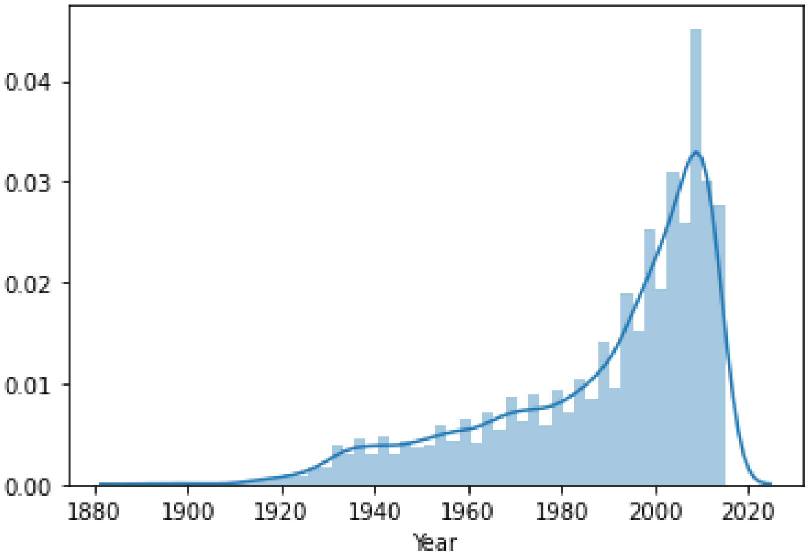


Figure 6.8: Movies released in each year

From the aboveFigure 6.8 we can see that most movies were released in 2010 compared with the number of movies released in 1940.

After that, we plot a correlation graph between all the genres by using the seaborn heatmap.

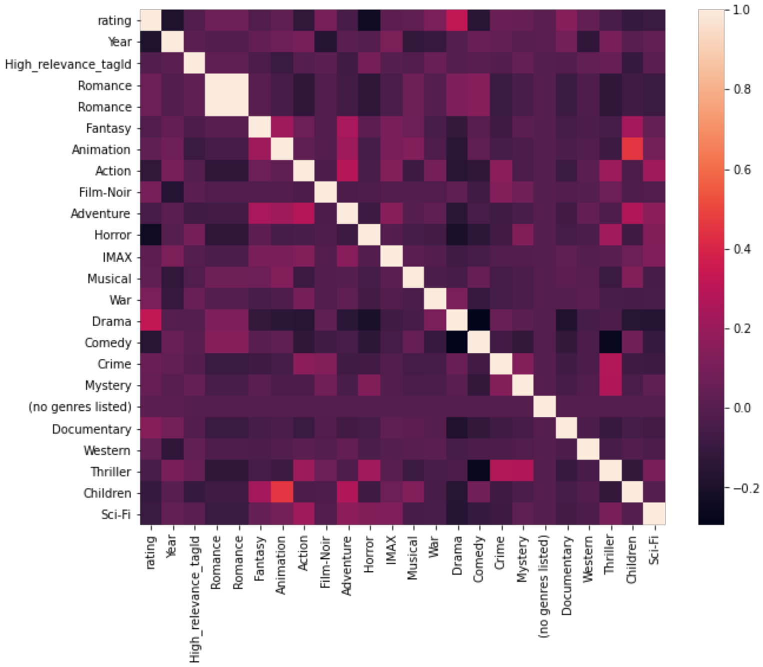


Figure 6.9: Correlation between all genres

For the above colormap Figure 6.9, we can see that there is a slight positive strength between Documentary and Children but a strong positive correlation between Animation and Children.

Chapter 7

Experiment

7.1 Result Analysis

As we are using the Federated Learning approach, our model improves over rounds. We divide our train data into five dummy clients to simulate the model. We train our model for 30 rounds, during which the global and local models transfer their learning and gradually improve. In both training and testing, model loss decreases over rounds shown in Figure 7.1

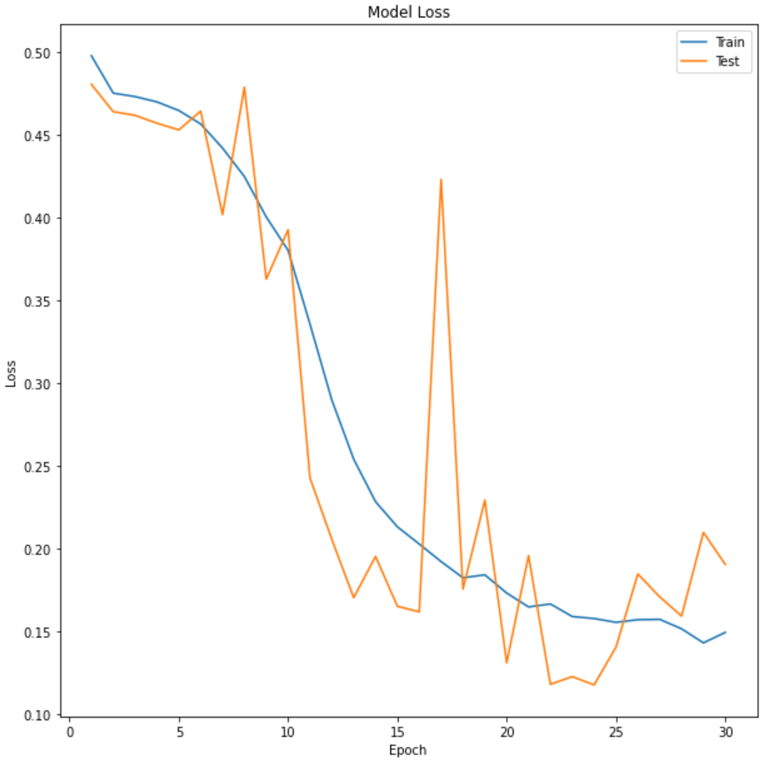


Figure 7.1: Model Loss

We chose Root Mean Square Error (RMSE) values for evaluation. The RMSE is used to evaluate regression-type models. This technique is widely used to evaluate the performance of recommendation algorithms. Basically, this is an error value, so the lower values are considered as better value. Additionally, we analyze the Mean

Absolute Error (MAE) of the client and server models. Brief evaluation of client and server models is shown in 7.1

Table 7.1: Experiment Result

	Client 1	Client 2	Client 3	Client 4	Client 5	Mean	Std
RMSE (testset)	0.8921	0.8776	0.8775	0.8874	0.8702	0.8810	0.0078
MAE (testset)	0.6541	0.6424	0.6376	0.6477	0.6347	0.6433	0.0070
Fit time	22.95	25.84	26.43	26.13	26.94	25.66	1.40
Test time	0.76	1.11	0.80	0.91	0.81	0.88	0.13
RMSE: 0.8906							

Computation Time: 1256.0954s

After 30 rounds, we obtain the root mean square error (RMSE), which is 0.8906. The learning and evaluation processes took a total of 1256 seconds to complete.

7.2 Interpretability of the Model

As our dataset is very large, for the sake of simplicity, we have built a similar Rating Matrix (R) to introduce neighbor style explainability. Applying equation 5.4 formula, to our hypothetically built Rating Matrix, we will have the table 7.2 where it represents the distance between every pair of user.

	User0	User1	User2	User3
User0	0.0	6.0	1.0	6.0
User1	6.0	0.0	5.0	2.0
User2	1.0	5.0	0.0	5.0
User3	6.0	2.0	5.0	0.0

Table 7.2: Neighbor Style Explainability Calculation

Table 7.3 presents, the level of impact having by other user while calculating the movie rating prediction for a particular user.

	Very Strong	Strong	Medium	Weak
User0	User0	User2	User1	User3
User1	User1	User3	User2	User0
User2	User2	User0	User1	User3
User3	User3	User1	User2	User0

Table 7.3: Explainability

Chapter 8

Conclusion

Responsible AI focuses on the ethical use of AI systems. Federated Learning ensures data privacy by sending only the model parameter to train the model. However, it is essential to ensure fairness and interpret the model to make the Federated Learning system more responsible. In our paper, we have proposed a Federated Recommendation system. To ensure the fairness of our system, we have used SDV++ for matrix factorization, which increases personalization. Besides, we have tried to introduce Explainable AI to our model to make the recommendation system more responsible. However, to our limitation, we could not compare our proposed model with other factor models such as SVD, Asymmetric-SVD in the FL recommendation system. We also believe there is ample scope to bring more interpretability to the model. In addition, we look forward to implementing the neighborhood model with SVD++ introduced in Koren's paper [1] in the FL recommendation setup to improve personalization more.

Bibliography

[1] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," Aug. 2008, pp. 426–434. DOI: 10.1145/1401890.1401944.

[2] F. Doshi-Velez and B. Kim, *Towards a rigorous science of interpretable machine learning*, 2017. arXiv: 1702.08608 `[stat.ML]`.

[3] F. D. Velez and B. Kim, *Towards a rigorous science of interpretable machine learning*, 2017. arXiv: 1702.08608 `[stat.ML]`.

[4] F. Chen, Z. Dong, Z. Li, and X. He, "Federated meta-learning for recommendation," *CoRR*, vol. abs/1802.07876, 2018. arXiv: 1802.07876. [Online]. Available: http://arxiv.org/abs/1802.07876.

[5] L. Gilpin, D. Bau, B. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," Oct. 2018, pp. 80–89. DOI: 10.1109/DSAA.2018.00018.

[6] D. Liu, T. Miller, R. Sayeed, and K. Mandl, *Fadl:federated-autonomous deep learning for distributed electronic health record*, Nov. 2018.

[7] M. Arivazhagan, V. Aggarwal, A. Singh, and S. Choudhary, *Federated learning with personalization layers*, Dec. 2019.

[8] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, *Explaining explanations: An overview of interpretability of machine learning*, 2019. arXiv: 1806.00069 `[cs.AI]`.

[9] G. Wang, *Interpret federated learning with shapley values*, May 2019.

[10] "Exploiting ai:how cybercriminals misuse and abuse ai and ml," 2020. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/exploiting-ai-how-cybercriminals-misuse-abuse-ai-and-ml.

[11] S. Lauritsen, M. Kristensen, M. Olsen, *et al.*, "Explainable artificial intelligence model to predict acute critical illness from electronic health records," *Nature Communications*, vol. 11, Jul. 2020. DOI: 10.1038/s41467-020-17431-x.

[12] P. Liang, T. Liu, L. Ziyin, R. Salakhutdinov, and L.-P. Morency, *Think locally, act globally: Federated learning with local and global representations*, Jan. 2020.

[13] Y. Mansour, M. Mohri, J. Ro, and A. Suresh, *Three approaches for personalization with applications to federated learning*, Feb. 2020.

[14] U. Pawar, D. O'Shea, S. Rea, and R. O'Reilly, "Explainable ai in healthcare," Jun. 2020. DOI: 10.1109/CyberSA49311.2020.9139655.

[15] F. Sattler, K.-R. Müller, and W. Samek, “Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–13, Aug. 2020. DOI: 10.1109/TNNLS.2020.3015958.

[16] S. Bag, “Federated learning – a beginners guide,” 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/federated-learning-a-beginners-guide/.

[17] D. Chai, L. Wang, L. Fu, J. Zhang, K. Chen, and Q. Yang, *Federated singular vector decomposition*, May 2021.

[18] B. Dickson, “What is federated learning?,” 2021. [Online]. Available: https://venturebeat.com/2021/08/13/what-is-federated-learning/.

[19] J. Fiosina, “Explainable federated learning for taxi travel time prediction,” Jan. 2021, pp. 670–677. DOI: 10.5220/0010485606700677.

[20] Y. Huang, L. Chu, Z. Zhou, *et al.*, “Personalized cross-silo federated learning on non-iid data,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 7865–7873, May 2021.

[21] J. C. Liu, J. Goetz, S. Sen, and A. Tewari, “Learning from others without sacrificing privacy: Simulation comparing centralized and federated machine learning on mobile health data,” *JMIR Mhealth Uhealth*, vol. 9, no. 3, e23728, Mar. 2021, ISSN: 2291-5222. DOI: 10.2196/23728. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/33783362.

[22] “Microsoft ai principles,” 2021. [Online]. Available: https://www.microsoft.com/en-us/ai/responsible-ai?activetab=pivot1%3aprimaryr6.

[23] K. Singhal, H. Sidahmed, Z. Garrett, S. Wu, K. Rush, and S. Prakash, *Federated reconstruction: Partially local federated learning*, Feb. 2021.

[24] “Flower: A friendly federated learning framework.” [Online]. Available: https://flower.dev/.