

Study of Machine Learning for Recommendation Systems

**Tushar Deshpandea, Khushi Chavanb, Ramchandra Mangrulkarc**

*Department of Computer Engineering, Dwarkadas J. Sanghvi college of Engineering, Mumbai, MS, India*

tushdeshpande791@gmail.coma, chavankhushi07@gmail.comb, Ramchandra.Mangrulkar@djsce.ac.inc

**Abstract**: This study provides an overview of recommendation systems and machine learning and their types. It outlines the types of machine learning such as supervised, unsupervised, semi-supervised learning, and reinforcement in brief. It explores how to implement recommendation systems using three types of filtering techniques: collaborative filtering, content-based filtering, and hybrid filtering. The machine learning techniques explained are clustering, co-clustering, and matrix factorization methods such as Single value decomposition (SVD) and Non-negative matrix factorization (NMF). It also discusses K-nearest neighbors (KNN), K-means clustering, Naive Bayes, and Random Forest algorithms. The evaluation of these algorithms is performed on the basis of three metric parameters: F1 measurement, Root mean squared error (RMSE) and Mean absolute error (MAE). For the experimentation, this study uses the BookCrossing dataset and compares analysis based on metric parameters. Finally, it also graphically depicts the metric parameters and shows what technique is the best and worst to incorporate into the recommendation system. This study will assist researchers in understanding the summary of machine learning in recommendation systems.

**Keywords:** Recommendation System, Machine Learning, Singular value decomposition (SVD), Non-negative Matrix factorization (NMF), Nearest k- neighbors (KNN), F1-measure, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE)

# INTRODUCTION

**Recommendation System**

The recommendation system [1] is the main part of digitization as it analyses the interest of users and recommends something based on those interests. The aim of these systems is to reduce information overload by retrieving the most similar items depending on the customer's interest. The primary use of these systems is decision making, maximizing profits, and reducing risks. This reduces customer’s efforts and time information searching. It works as a filter that suggests alternatives based on massive data. Moreover, it acts as a multiplier that contributes to the expansion of the client’s options.

Over the last few years, the enthusiasm for recommendation systems has increased tremendously. This is the most widely used service on high-end websites like Amazon, Google, YouTube, Netflix, IMDb, TripAdvisor, Kindle, etc. A number of media companies develop these systems as a service model to their clients. Furthermore, the implementation of such systems at commercial and non-profit sites attracts the attention of the customer. These also satisfy clients more with online research results. These systems help customers search for their loved items faster and acquire more authentic predictions leading to higher sales at an eCommerce site.

About knowledge of these systems, there are variety of undergraduate and graduate courses at institutions around the world. Conferences, workshops, and contests are organized in accordance with these systems. One of the competitions was the Netflix Prize, which was organized around machine learning and data mining. In this competition, participants were required to develop a movie recommendation system whose accuracy is 10% more precise than that of the existing system, also known as Cinematch. After a year of hard work, the Korbell team won first place using the two main algorithms: matrix factorization (Singular value decomposition (SVD)) and Restricted Boltzmann machines (RBM).

Real applications [2] employ different ML algorithms such as K-nearest neighbor (KNN), Naive Bayes, Random Forest, Adaboost, Singular value decomposition (SVD), and many others. The evolution of the recommendation scheme has led to the application of ML and AI algorithms for effective prediction and accuracy. In addition, the results provided by some ML algorithms are expected to be slightly promising. Because of the broad classification of ML algorithms, the choice of an ML algorithm may become a challenge depending on the different situations where recommendation systems are needed. To select an effective ML algorithm, the best way for the researcher or programmer would be to have a thorough knowledge of ML and recommending systems. This knowledge enables the researcher to create a model appropriate to a specific problem. Here, the study provides an overview of ML briefly.

# Machine Learning

Machine learning demonstrates the imitation of human learning in computers by learning from experiences and applying them to recently encountered situations. ML originated in the 1950s but became more popular in the 1990s. Humans understand but on the other side, computer uses algorithms.

Machine Learning is classified into four categories:

1. Supervised learning
2. Semi-supervised learning
3. Unsupervised learning
4. Reinforcement learning

## Supervised learning

This learning deals with algorithms that provide training data with a set of features and the correct prediction according to those features. The task of the model would be to learn from this data and apply the information learned into new data with the input features and predict its outcome. One example where supervised learning is used is predicting the price of a house based on a given set of features. This may include the size of the house, its location, its floor, etc. The data provided are these features and on the basis of those features, the output which is the price is predicted. For instance, the price of an apartment on a higher level and close to the airport would be higher than the price of a home in a remote area.

## Semi-supervised learning

In this learning, the model learns from training data that includes missing information. These types of algorithms focus more on concluding from insufficient data. An example of semi-supervised learning would be evaluation of movies through its reviews. The reason this falls under semi-supervised learning is because not all the viewers of the film will give their review, so we've only partly labeled the data. The features considered could be the name of reviewer, movie title, general rating, critics rating, and the movie’s payout. Here, the general rating feature can have a few missing values as not every person will give his or her rating.

## Unsupervised learning

This learning focuses on algorithms that do not require training data. These algorithms use real-world information to learn by themselves. It focuses primarily on relations hidden in the specified data. A use case of unsupervised learning can be a movie-based recommendation system where the features considered can be the movie’s title, its genre, and overall rating. For example, if Sarah watches movie A and movie B of the same genre and John watches movie A then our model should give movie B as one of the recommendations to John. This type will be discussed in detail later in this chapter.

## Reinforcement learning

This type of learning involves algorithms that learn from feedback from an external body. It is similar to a student and teacher where the teacher may give fewer grades (negative feedback) or more grades (positive feedback). An example is to offer a treat to a dog for a positive response and not give that treat for a negative one.

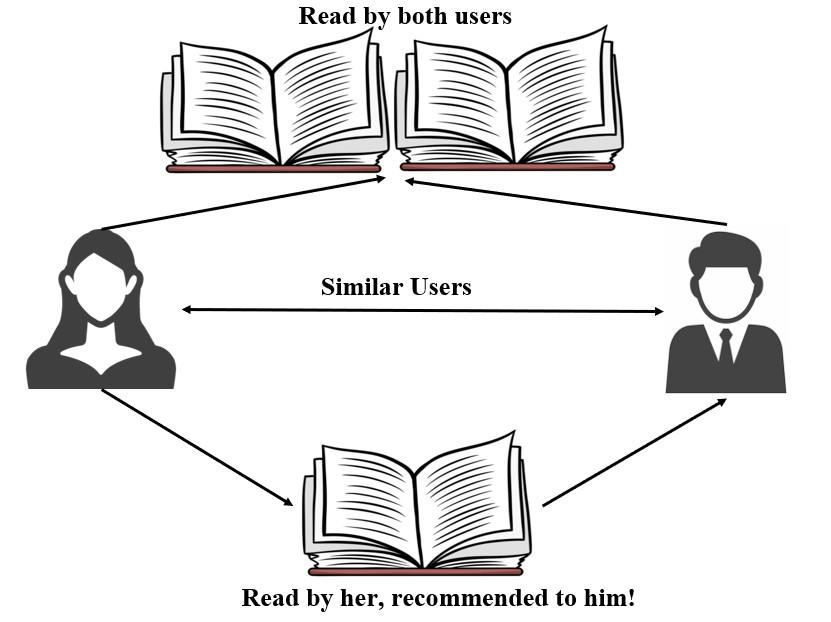
# METHODS

The idea of recommendation systems is to provide recommendations to the user according to their behavior or profile. It analyzes the user's interest dynamically so that when the user carries out actions, he recommends according to his tastes. Various types of recommendations also involve recommendations based on trust, context, and risk. The types discussed in this document can be found in Fig.6. The Recommendation System [4] is mainly divided into three categories:

1. Collaborative filtering
2. Content-based filtering
3. Hybrid filtering

# Collaborative filtering

In this approach [5], recommendation systems work according to user information. It compares users of similar preferences and recommends trying items that other users have tried. An example is book applications in which the model would search for similar preferred users and would recommend what was purchased by those users to the current user. This type of system is further divided into a memory-based and model-based approach.



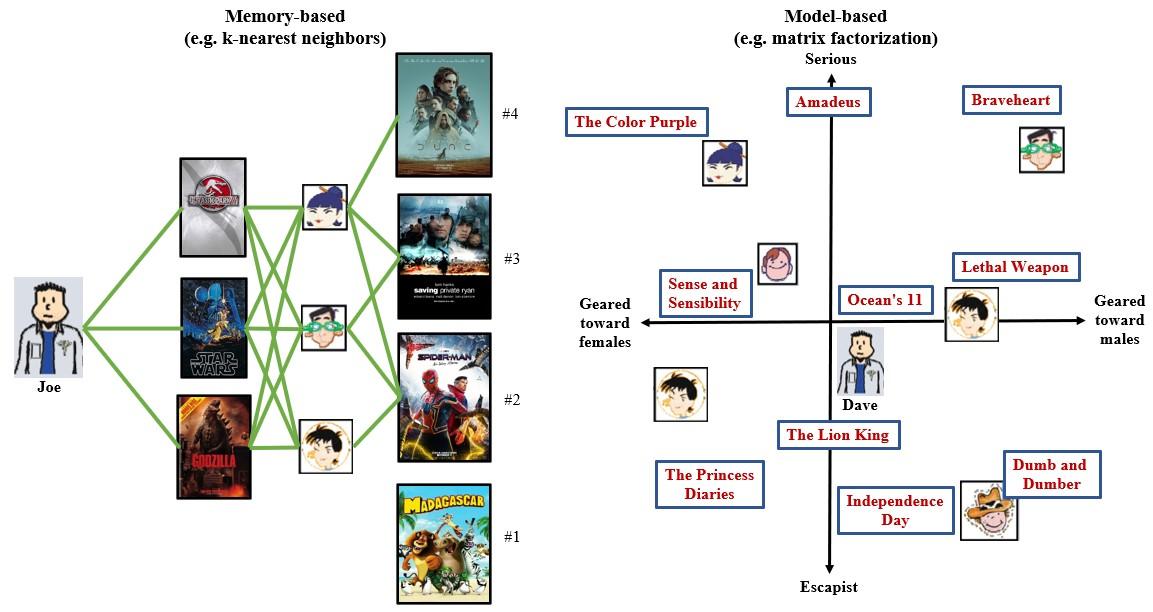
**Figure 1:** Example of collaborative filtering [6]

## Model-Based

In this method [7], the information base is past evaluations by which the model learns for better future predictions. This method functions on items that are not yet seen or used by the user. This method increases the accuracy of the system. Model-based approaches include matrix factorization, clustering, association techniques, Bayesian networks, and many more.

## Memory-Based

In this method, the basis of the information is the likes and dislikes of other users, which is similar to the profile of the user who requires recommendations. This approach analyses the similarity between user interests to predict an item to the desired user. The approach is divided into subtypes, particularly user-based and item-based methods.



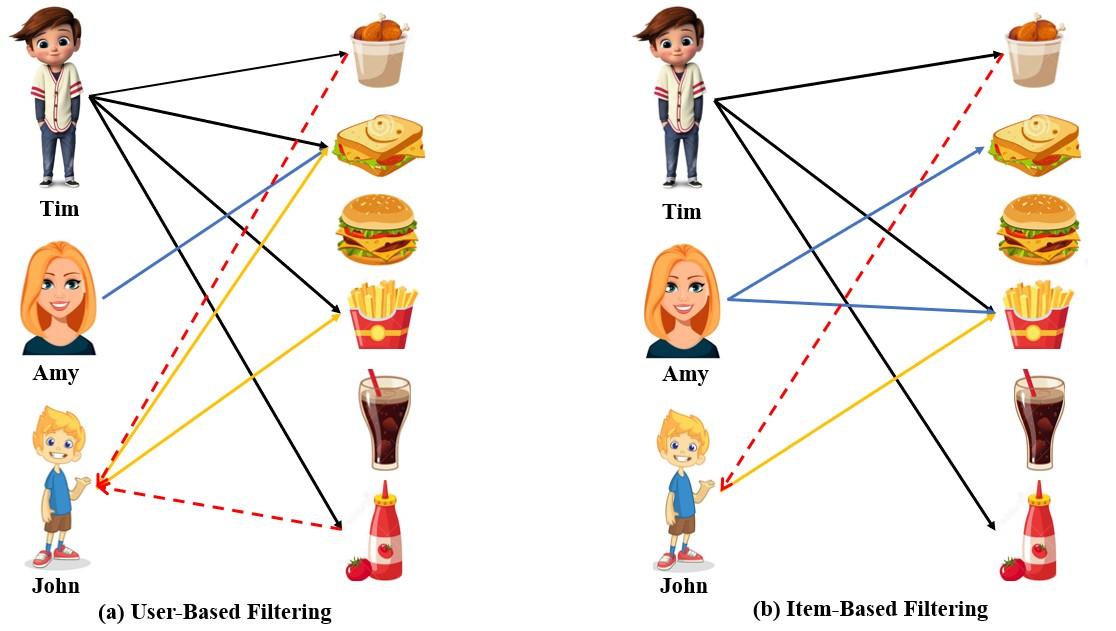
**Figure 2:** Difference between memory-based [8] and model-based [9]

## User-Based

This approach analyses the similarity among users in predictions. It can also predict, depending on the desired user's behavioral patterns. For example, if a user purchases a book, they will analyze other users' preferences on that book and recommend the new items to the user.

## Item-Based

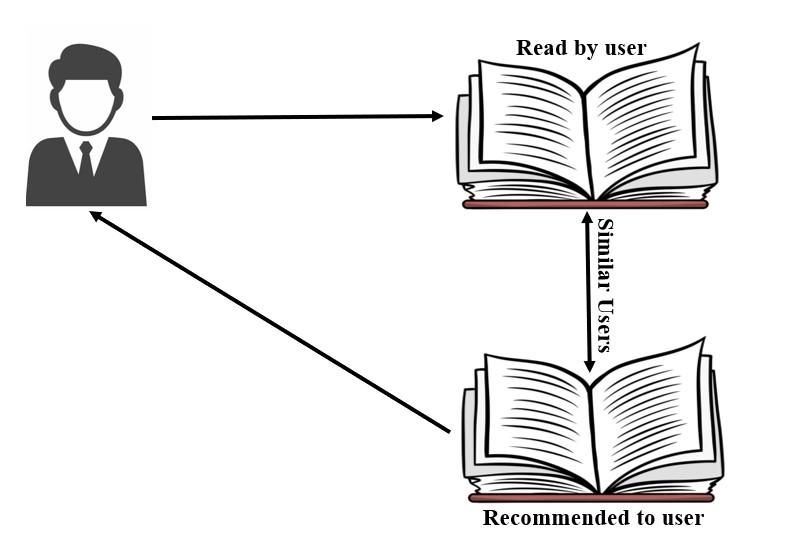
This approach analyzes the similarity between the items researched or purchased by users for predictions. In other words, it computes the similarities between items unknown to the user and items known to the user and displays unknown items if the similarity value is high. For example, if a user buys an item, this system will look for items with similar features to the item purchased and recommend it to the user.



**Figure 3:** Difference between user-based and item-based [10]

# Content-based filtering

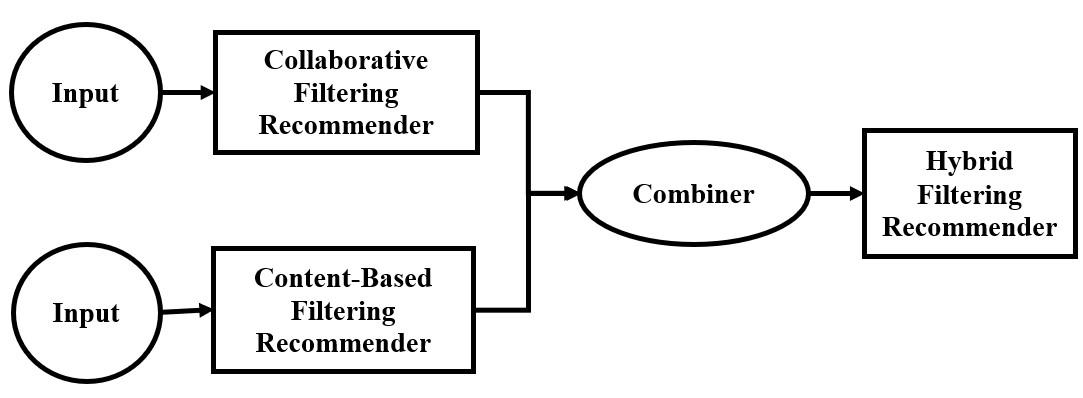
In this approach, the recommendation system functions based on the data of the item the user is looking for. The model analyses other items with attributes similar to those in the search and recommends them to the user. An example is online shopping, where the user searches for an item with specific features and recommends similar items.

****

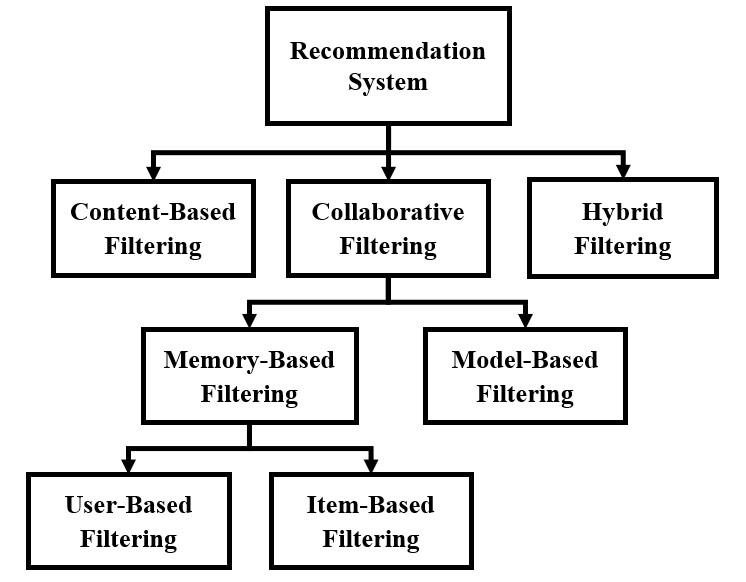
**Figure 4:** Example of content-based filtering [6]

# Hybrid filtering

This approach is a combination of the two earlier methods, as illustrated in Fig. 5. This means that these recommendation systems are based on item data and user information. The first step consists of analyzing the user information. The second step is to analyze the data element you are looking for or using. Finally, the relevant dataset of the first two steps appears in the form of recommendations.



**Figure 5:** Mechanism of Hybrid filtering

****

**Figure 6:** Tree diagram of Filtering Techniques

# Algorithms

This article includes a detailed explanation of Singular value decomposition (SVD), Non-negative matrix factorization (NMF), K-means clustering, K-nearest neighbors (KNN), Co-clustering, Naive Bayes, and Random Forest algorithms.

## Co-clustering

Co-clustering, also known as bi-clustering [11], is a method wherein there is a simultaneous clustering between rows and columns of a matrix. This matrix represents information as a function of user characteristics and item characteristics. In other words, co-clustering can also be visualized as grouping two different kinds of entities according to their similarity. The result of a co-clustering algorithm is commonly termed a bi-cluster [12][13]. The kinds of bi-clustering are classified according to the nature of these bi-clusters. It depends mainly upon constant and consistent values.

1. Bi-cluster with constant values: Rows and columns within a clustering block have the same constant value.
2. Bi-cluster with constant values in rows or columns: Every row or column in a clustering block has the same constant value.
3. Bi-cluster with coherent values: These bi-clusters identify more complex similarities between genes and conditions using an additive or multiplicative method.

It is used across a wide variety of applications. Rege et al. [14] use co-clustering for clustering documents and topics. Chen et al. [15] and Felzenszwalb and Huttenlocher[16] use image co-clustering for image processing. It also helps to identify interaction networks [17][18]. It is also an analytical tool for election data. The clustering technique is implemented through a variety of matrix factorization techniques.

## Matrix Factorization

Matrix factorization is a type of algorithm associated with the decomposition of the user-item interaction matrix into the product of two rectangular matrices. This is usually done by minimizing the mathematical cost function RMSE (Root mean square error) which is done using gradient descent. Because of its effectiveness, this method became more popular during the Netflix Prize challenge (as discussed above). Recommendation systems use different matrix factorization techniques. Furthermore, a detailed study on Singular value decomposition (SVD) and Non-negative matrix factorization (NMF) is given below.

## Singular Value Decomposition

This method is associated with linear algebra and is increasingly popular within ML algorithms. Its application is mainly recommendation systems for e-commerce, music, or video streaming sites.

SVD refers to the decomposition of a single matrix into three additional matrices. The general form is: 𝑀 = 𝑋𝑆𝑌T  (1)[19]

where M is the given mxn matrix,

X is an mxn orthogonal matrix that denotes the relation between the user and latent factors,

S is an nxn diagonal matrix that denotes the strength of these latent factors, and

Y is nxn orthogonal matrix and it represents the similarity between the user and latent factors.

The steps involved in SVD are given below.

1. In the first step, the data is represented as a matrix with rows as user and columns as items.
2. If there are any empty entries in the matrix, provide the average of the other entries so that there is no major error in the calculation.
3. After this, compute the SVD. (Done using NumPy and Surprise library)
4. After calculating the SVD, you only need to reduce it to obtain the expected matrix that will be used for the prediction by looking at the appropriate user/article pair.

The primary benefit of SVD is that it simplifies the data set and eliminates noise from the data set. It also functions with the numerical data set. Also, it could improve the precision. There are many issues related to the SVD. One of the most important issue is data scarcity, also called the cold start problem [20]. This occurs due to a new community, user, or item. If a new community, user, or item is added, the recommendation system will not work properly due to a lack of information. Black sheep is also an issue, meaning some customers also agree and disagree with the same group of people. If so, it is impossible to make recommendations. Due to its temporal complexity (O (n)), it also suffers from scalability issues.

There are different applications of SVD. The most common applications are pseudo- inverse, resolving homogeneous linear equations, minimizing total least squares, range, null space and rank, and approximation to the lowest rank matrix. In addition, it is used for signal processing, image processing, and big data.

## Non-negative Matrix Factorization

This is also a matrix factorization technique [21]. As with SVD, the analogy for this approach is to break down or factorize a given matrix. The only difference, on the other hand, is that the matrix is split into two parts. The two parts are called W and H. W matrix is for weights which represent each column as a basic element. These are building blocks from which to obtain predictions to the original data item. H matrix is hidden which represents the coordinates of the data items of W. In other words, it guides us in converting to the original data item from the group of building blocks of W.

The order of execution in NMF is given below.

1. Import the NMF model using the Surprise library.
2. Then, load the dataset and isolate it to the given model.
3. Later, clean the data and create a function to pre-process data.
4. Successively create a document term matrix 'V'(given matrix).
5. Create a function to display the mode features.
6. Then, run NMF on the document term matrix 'V'.
7. Continue checking and iterating until useful features are found.

The advantage of NMF is that it breaks down the given matrix into two smaller matrices whose dimensions can be controlled by the given matrix. It differs from other matrix factorization algorithms because it works only on positive numbers which makes the data interpretable. The dataset can become smaller if W and H are depicted sparsely. The issue with the semi-supervised NMF is that depending on the number of data points available, there is a reduction in the fitted data points.

Applications of the NMF include the processing of audio spectrograms, document clustering, recommendation systems, chemometry, and many others. It is also used for dimensionality reduction in astronomy, statistical data imputation, as well as nuclear imaging.

## Difference between SVD and NMF

So as stated above, both SVD and NMF are matrix factorization techniques. But there are also some differences between them, which could help us to choose the best algorithm for a situation between these two.

1. The SVD includes both negative and positive values, while the NMF has strictly positive values. That makes NMF useful because it provides more sense and connections are made easier.
2. SVD factors can be related to the eigen functions of a system where the original matrix denotes a system about which one is taking interest from a signal processing perspective. This makes SVD more effortless. Although NMF can also be used for the same purpose because the association is indirect in this approach, it becomes more tedious.
3. The factors of SVD are unique whereas the factors of NMF are not unique. As a result, NMF is better for algorithms with privacy protection.
4. SVD factors into three matrices out of which the sigma matrix gives the information stored in the vector. Whereas NMF only factors into two matrices which do not include the sigma matrix.

## K-Nearest Neighbors

KNN is an easy machine learning algorithm based on supervised ML learning. It finds similar items based on the distance between test data and individual training data using a variety of distance concepts. In this algorithm, predictions are mainly made using the calculation of the Euclidean distance of the nearest neighbors. Besides, the use of Jaccard similarity, Minkowski, Manhattan, or Hamming distance can be done instead of Euclidean. This is a non-parametric algorithm that assumes nothing on the given data. It is also referred to as a lazy learning algorithm, which does not learn from data, instead stores and performs actions on the data.

The steps involved in KNN are given below.

1. Load the dataset and preprocess it.
2. Fit the KNN algorithm (defined as Nearest-Neighbors) to the training dataset (use the scikit-learn library). For using the Surprise library, it is defined as KNNBasic.
3. Predict the test result.
4. Creating the confusion matrix and finding the test accuracy of the result.
5. After this, the visualization of the test result can be done.

This algorithm is used as it is easy to interpret the result. It also has great predictive power and less computing time. The main issue with KNN is that it becomes much slower as the volume of data increases. As such, it does not give good accuracy with large datasets. It is also highly sensitive to missing values, outliers, and noise from the dataset.

It is primarily used for classification and regression problems. The result of a classification problem is a discrete value while for a regression problem, the result is a real number (containing a decimal). It is commonly used for text extraction. It is used in finance for stock prediction, management of loans, and analysis of money laundering. It is used in agriculture for weather forecasting and estimation of soil water parameters. It is also used in medicine to predict different diseases.

## K-means clustering

The k-means algorithm is the most widely known clustering algorithm. It is the simplest method of unsupervised learning to resolve the clustering issue. It also aims at solving the Expectation-Maximization problem. In this algorithm, a k value is received that represents the number of clusters. Then it classifies the data set by dividing it into a given number of clusters of similar characteristics/preferences. The similarity is calculated using the distance of two items. In this method, the distance is measured using a square Euclidean, Manhattan, Euclidean, or Cosine distance measure. This method is evaluated using the elbow method or silhouette analysis.

Euclidean: 𝑑(𝑥, 𝑦) = (2)[22]

Squared Euclidean: 𝑑2(𝑥, 𝑦) = (𝑦2 − 𝑥2)2 + (𝑦1 − 𝑥1)2 (3)[23]

Manhattan: 𝑑(𝑥, 𝑦) = |𝑥1 − 𝑦1| + |𝑥2 − 𝑦2| (4)[24]

Cosine distance:

where x1, y1, x2, y2 are the coordinates of the data points and (𝑝, 𝜃) and (𝑞, 𝜓) are the polar coordinates of x and y.

## Naive Bayes

Naive Bayes [3] is an ML probabilistic algorithm that is based on the Bayes theorem. Such algorithms result in each pair of items or features being independent of each other. In Naive Bayes, the assumptions are that each feature provides an independent and equal part in the outcome. To start, the Bayes theorem is discussed below.

P(𝑋⁄𝑌) = P(𝑌⁄𝑋) ∗ 𝑃(𝑋)⁄𝑃(𝑌), (6)[26]

where P(𝑋⁄𝑌) is the probability of X given that Y event has occurred, P(𝑌⁄𝑋) is the probability of Y given that X event has occurred,

P(𝑋) is the probability of event X*,* and

P(𝑌) is the probability of event Y.

The types of naive Bayes are: Bernoulli, Multinomial, and Gaussian naive Bayes.

Bernoulli naive Bayes: This is a binary algorithm that interprets whether a feature is present or not. It is used when there are binary function vectors (i.e., ones and zeroes). One of its applications is the bag of words model for text classification.

It follows the following rule:

𝑃(𝑥i⁄𝑦) = 𝑃(𝑖⁄𝑦) ∗ 𝑥i + (1 − 𝑃(𝑖⁄𝑦))(1 − 𝑥i) (7)[27]

where x and y are two events and i is a subevent of x.

Multinomial naive Bayes: Feature vector refers to the frequencies that are made using the multinomial distribution. It is used efficiently for working with texts in natural language processing.

Gaussian naive Bayes: Values associated with each feature vector are generated by Gaussian distribution or Normal distribution. If this is shown graphically, it results in a bell-shaped curve. The equation for this is as follows:

The steps involved in naive Bayes are written below:

1. The dataset is first preprocessed.
2. The fitting of Naive Bayes in the training data
3. Predict the features of the test data.
4. Create the confusion matrix and get the accuracy of the model
5. Try to visualize the result of the testing set.

The advantage of naive Bayes is that it is quick and precise for predictions. Such an approach also reduces the complexity of the computations. It can be used not only for one but also for problems with multiple feature classes. This algorithm works best if the variables are discrete and not continuous. The main disadvantage of naive Bayes is the assumption that features are independent of each other, which is not possible in real life. Moreover, if there is no training set for a particular class feature, this may result in a posterior probability of zero. This is known as the zero-frequency problem.

There are a variety of applications of naive Bayes. A major application of Naive Bayes lies in the recommendation system. If collaborative filtering and naive Bayes both are integrated into the recommendation system, it can predict through the unseen information regardless of preferences. As well, text classification is a popular application of naive Bayes. Applications of naive Bayes are real-time predictions and multiclass predictions for classification problems. It can also be used for facial recognition, medical testing, and weather forecasting.

## Random forest

The random forest algorithm [29] is a common supervised machine learning technique based on the ensemble learning concept. Ensemble learning is a method of combining various classifiers to improve model accuracy. In this algorithm, the dataset is split into several subsets and then contained in the same number of decision trees. Instead of depending on a decision tree, this algorithm takes an average of the predictions of all decision trees. This makes the outcome of the predictions more accurate.

Steps involved in implementing a random forest algorithm are given below.

1. The dataset is loaded and then preprocessed by splitting the data into a training and testing set.
2. The training and testing data are then feature scaled.
3. The training set is used to fit the random forest algorithm (defined as RandomForestClassifier). This is done by importing the scikit-learn library.
4. Prediction of the test result is done using a new prediction vector.
5. To conclude, a confusion matrix is created. This matrix gives the correct and incorrect predictions.
6. Visualization of the test result is done.

The main advantage of this algorithm is its versatility. It has increased predictability. So, this is a handy algorithm to use. It also overcomes the biggest problem of overfitting. It can handle a large dataset and also needs less time to train the dataset. The major drawback is that many decision trees can delay the algorithm and not function efficiently in the real world. It is used for both classification and regression, although it is not appropriate for regression.

There are various application domains of the random forest method. In banking, it is used for fraud detection, loan risk identification, and various identifications and detections are performed based on banking services. In medicine, it is used to find the combination of medications and also to predict the risk and patterns of the disease. In commercialization, it can be used to predict stock prices and trends. It is also used in satellite imagery and object and multiclass detection.

# Evaluation Methods

There are various methods used in the evaluation of machine learning methods. One of the commonly used methods is the absolute error and accuracy-based evaluation methods such as RMSE (Root mean squared Error), MSE (Mean square error), and MAE (Mean absolute error). There are decision support methods like precision, recall, F1-measure, and ROC (Receiver operating characteristic) curve. In addition, there are ranking-based evaluation methods, such as nDCG (Normalization of discounted cumulative gain), MRR (Mean reciprocal rank), mean precision, and Spearman rank correlation. Moreover, different metric evaluation methods assess performance based on prediction, decision, and ranking power. Examples of these metric-based approaches include coverage, popularity, novelty, diversity, and temporal evaluation. Finally, business sector metrics can be used to reach its objective. The above- mentioned algorithms will be evaluated using F1-measure, RMSE, and MAE.

## F1 measure

This accuracy measurement combines accuracy and recall and is also called the harmonic average of the model. This is used to measure the accuracy of the model.

The formula for the F1 measure is F1=2\*P\*R/(P+R), where P and R are the precision and recall of the model.

Precision: This measure, also known as the TP (True positives), is defined as the ratio of TP to the sum of TP and FP (False positives).

Recall: This measure, also known as sensitivity, is defined as the ratio of the TP to the sum of TP and FN (False negatives).

To avoid the least robustness of normal accuracy measurements, this measurement is preferred since it can take note of variations of different types of errors. The F1 measure is efficient whenever there is a presence of different costs of FP(False positives) and FN(False negatives). The F1 measurement can also be useful if there is an imbalance in the class feature numbers because in such cases the precision can be very misleading. The weakness of the F1 measurement is that the value calculated for one feature is independent of the other. In other words, it cannot compute the effectiveness of two features combined or based on each other's information. The applications for the F1 measurement include information retrieval in NLP (Natural Language Processing). This is most frequently used in search engine systems. In addition, it is most commonly used in binary classification systems.

## RMSE (Root Mean Squared Error)

It is a performance measure of the ML models that are primarily calculated to see how well the model fits (i.e., less error, more accuracy). In other words, this is used to predict quantitative data. It is defined as:

In the above RMSE equation, 𝑦j is the original data and is the predicted data.

This measure is used because it is quite easy to distinguish. This makes it easier to work with methods such as gradient descent. This is also good for evaluating the standard deviation for distributing the errors generated.

RMSE has square errors, so even a small error can affect the value immensely which allows us to ensure that the model yields as little error as possible. This means that an error of 10 will become 100 times worse than an error of 1. RMSE could become difficult to understand from an interpretation point of view as it contains square values whereas MAE would be clear to understand due to absolute values.

## MAE (Mean Absolute Error)

This measure is also used as an alternative to RMSE. MAE is the average of the absolute difference between the original data and the predicted data. If this absolute value is not taken, this would become the mean bias error (MBE).

To represent MAE mathematically:

In the above MAE equation, is the original data and is the predicted data. MAE is more stable than RMSE when the variation in frequency error distribution increases. This means that an error of 10 will be 10 times worse than an error of 1. MAE is generally preferable when scales of error are linear whereas RMSE is preferable when scales of error are non-linear. MAE is not useful when no absolute value is required, in such cases RMSE is preferable.

| MAE | RMSE |
| --- | --- |
| It doesn’t consider the sign of the input, if the input is negative it takes the positive value. | It considers the sign of the input whether it is positive or negative. |
| It is less biased towards large values. Thus, when it comes to a large error, it does not reflect the result of the algorithm. | When it comes to large errors, it reflects in the result of the algorithm. Thus, it is much better than MAE. |
| The MAE value is comparatively smaller as the sample size increases. | RMSE is comparatively higher than MAE for increasing sample size. |
| MAE restricts larger errors. | RMSE does not restrict large errors. |
| MAE is preferred where there is a proportion between overall performance and an increase in error. | RMSE is preferred where the overall performance and the increase in error are disproportionate. |

**TABLE 1:** Difference between RMSE and MAE [33]

# EXPERIMENTATION

# Dataset

The BookCrossing dataset [34] is built by CAI-Nicolas Ziegler from Amazon Web Services. There are 270,000 books read by 90,000 users with 1.1 million reviews. The data consist of three tables which include information about ratings, books, and users. This data is downloaded from Kaggle. The rating dataset provides a list of book ratings given by the users. It includes 1,149,780 rating records containing 3 fields: userID, ISBN, and bookRating. The ratings are either explicitly expressed on a scale of 1 to 10 or implicitly expressed by zero. As shown in Fig. 7, the vast majority of ratings are 0 and these ratings are distributed very unevenly. The books dataset provides book information, which includes 271,360 book records containing 8 fields. First, 5 fields containing the content-based information: ISBN, Book-Title, Book-Author, Year-Of- Publication, Publisher, and the last 3 image-URL fields: Image-URL-S, Image-URL- M, Image-URL-L. These 3 different URL images are linked to the cover page of the books according to their size. The user dataset provides demographic information of users. It includes 278,858 user records and 3 fields: user id, Location, and Age. Fig.8 shows that the majority of active users are youth between the ages of 20 and 30.

Chart, histogram

Description automatically generated

**Figure 7:** Rating Distribution of the books in datasetChart, histogram

Description automatically generated

**Figure 8:** Age Distribution of users in user-data

# Libraries used

# *NumPy*

# NumPy is a fundamental package and linear algebra library that is an essential tool for scientific computations & mathematical operations in python. It is a key component of research analysis pipelines in a wide range of fields such as physics, chemistry, astronomy, geology, biology, psychology, materials science, engineering, finance, and economics. For example, in astronomy, NumPy was an integral part of the discovery of gravitational waves as well as the first imaging of a black hole.

# NumPy is also an open-source, community-developed library, which provides multidimensional arrays along with array-aware functions. Since NumPy array is simple and easy to use, it is the de-facto format for displaying array data in Python.

# *Surprise*

# Surprise is an easy-to-use Python scikit for recommender systems. It utilizes built-in Python data structures (primarily dictionaries) and NumPy arrays. It provides a collection of estimation algorithms for rating prediction. It also includes the implementation of classical algorithms such as the main similarity-based algorithms, and matrix factorization algorithms like SVD or NMF. Moreover, it supports tools for model evaluation, including cross-validation iterators, built-in metrics, and tools for selecting models and searching for hyperparameters, such as grid search and randomized search. Lastly, it was designed to be helpful to researchers who want to quickly explore new recommendation ideas through the creation of custom prediction algorithms.

# *Scikit-learn*

# Scikit-learn is a Python module that integrates a wide variety of ultra-modern computer learning algorithms for solving medium-scale supervised and unsupervised problems. This bundle aims to make the knowledge of machine learning easily accessible to non-specialists using a general-purpose high-level language. The scikit-learn library provides several efficient tools for machine learning and statistical modeling such as classification, regression, clustering, and dimensionality reduction. It is used to build machine learning models instead of reading the data, manipulating, and summarizing it.

# Implementation

The book recommendation system has been done using item-based and user-based collaborative filtering experimented in python and compiled in Jupyter Notebook. After evaluating the RMSE scores of the user and the item, optimization of the book recommendation system is done by integrating various other algorithms such as co- clustering, SVD, NMF, KNNbasic, KNNwithMeans, and KNNwithZScore models from the Surprise library.

# Result

In this article the BookCrossing Dataset was implemented. It contains three tables. One table contains the user's information. The second table includes information on books. The final table includes the book routing information. Experimentation employed user-based and item-based collaborative filtering methods for the desired recommendation system. The RMSE score of these methods varied from 7 to 8. For improvement, the use of co-clustering, SVD, NMF, KNNbasic, KNNwithMeans, and KNNwithZScore models is done. The use of the above algorithms allowed a dramatic improvement of RMSE and MAE errors. The following table shows the RMSE value, the MAE value, and the F1 score for the implemented algorithms.

| Algorithm | RMSE | F1 measure | MAE |
| --- | --- | --- | --- |
| Co-clustering | 1.8393 | 0.4289 | 1.4274 |
| SVD | 1.5726 | 0.4428 | 1.2046 |
| NMF | 2.4767 | 0.4202 | 2.0717 |
| KNNBasic | 1.9473 | 0.4434 | 1.5263 |
| KNNwithMeans | 1.7994 | 0.4404 | 1.3925 |
| KNNwithZScore | 1.7967 | 0.4402 | 1.3819 |

**TABLE 2:** RMSE, MAE, and F1 measure of algorithms implemented on dataset.

# Discussion

By comparing the values obtained from the above analysis, the graphic display is shown below in Fig. 9 and Fig. 10. For a suitable algorithm, it is necessary to use a smaller RMSE and MAE measurement and a greater F1 measurement. Fig. 9. also represented that RMSE is higher than MAE. This is due to the differences mentioned earlier in Table 1. Thus, in the measure of errors, for comparison, the RMSE value is much better than the MAE value. As well, the F1 measurement is used for the confusion matrix.

Chart, line chart

Description automatically generated

**Figure 9:** RMSE and MAE comparison of implemented algorithms

(Error(%) on y-axis and Algorithms on x-axis)

Chart, line chart

Description automatically generated

**Figure 10:** F1 measure of implemented algorithms

(F1-measure(%) on y-axis and Algorithms on x-axis)

# CONCLUSION

Hence, this work concludes that the SVD technique is the most preferred among the algorithms implemented. Fig. 9 shows that the NMF has a large RMSE and MAE and less F1 measurement compared to others. It further concludes that the NMF alone is not suitable for this dataset. Moreover, KNN (includes KNNBasic, KNNwithMeans, KNNwithZScore) is much better compared with NMF primarily based on RMSE and MAE values. In addition, it concludes that the evaluation of the RMSE is much better than that of the MAE.

# ACKNOWLEDGEMENT

We would like to express our gratitude to Department of Computer Engineering at Dwarkadas J Sanghvi College of Engineering who motivated us to dive into research and guided us when we faced any difficulty. Also, the assistance provided by our senior classmate Onkar Thorat is greatly appreciated.

**Consent of publication:**

The paper is not published in any other publications. The table(s)/figure(s) which is/are not cited is/are original, have not been published before, or is in the public domain.

**Conflict of interest:**

There is no conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

# References

1. Silveira, Thiago & Zhang, Min & Lin, Xiao & Liu, Yiqun & Ma, Shaoping, “How good your recommender system is?” A survey on evaluations in recommendation. International Journal of Machine Learning and Cybernetics. 10. 10.1007/s13042-017-0762-9, 2019.
2. Shani, Guy & Gunawardana, Asela, “Tutorial on application-oriented evaluation of recommendation systems”, Ai Communications. 26. 225-236. 10.3233/AIC-130551, 2013.
3. Nguyen, Sang,“Model-Based Book Recommender Systems using Naïve Bayes enhanced with Optimal Feature Selection”, 217-222. 10.1145/3316615.3316727, 2019.
4. F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh, “Recommendation systems: Principles, methods and evaluation”, Egyptian Informatics Journal, Volume 16, Issue 3, 2015, Pages 261-273, ISSN 1110-8665, <https://doi.org/10.1016/j.eij.2015.06.005>.
5. P. Valdiviezo-Diaz, F. Ortega, E. Cobos and R. Lara-Cabrera, "A Collaborative Filtering Approach Based on Naïve Bayes Classifier," in IEEE Access, vol. 7, pp. 108581-108592, doi: 10.1109/ACCESS.2019.2933048, 2019.
6. Sanket Doshi, “Brief on recommender systems”, Towards Data Science, Feb 2019, [https://miro.medium.com/max/1064/1\*mz9tzP1LjPBhmiWXeHyQkQ.png](https://miro.medium.com/max/1064/1*mz9tzP1LjPBhmiWXeHyQkQ.png)
7. Do, Minh-Phung & Nguyen, Dung & of Loc Nguyen,“Model-based approach for Collaborative Filtering”, Academic Network. 2010.
8. Ayang Laishram, “Novelty in Recommender Systems”, Towards Data Science, Nov. 2019, https://miro.medium.com/max/449/0\*IU9e4BZiaflPb\_iL.png
9. Will Johnson, “Recommender Systems with Apache Spark’s ALS function”, Slideshare.net, May 2016, https://image.slidesharecdn.com/20160503mkebdrecosys-160501205055/95/recommender-systems-with-apache-sparks-als-function-10-638.jpg?cb=1462136016
10. Ayse Yaman, CodeX, “Hybrid Recommender System-Netflix Prize Dataset”, Medium, Retrieved from https://miro.medium.com/max/1370/0\*PCZeW5TphSgtkIqm.png
11. Beatriz Pontes, Raúl Giráldez, Jesús S. Aguilar-Ruiz, “Biclustering on expression data: A review”, Journal of Biomedical Informatics, Volume 57, Pages 163-180, ISSN 1532-0464, <https://doi.org/10.1016/j.jbi.2015.06.028>, 2015.
12. Gan, Xiangchao & Liew, Alan Wee-Chung & Yan, Hong, “Discovering biclusters in gene expression data based on high-dimensional linear geometries”, BMC bioinformatics. 9. 209. 10.1186/1471-2105-9-209, 2008.
13. Das, Joydeep & Mukherjee, Partha & Majumder, Subhashis & Gupta, Prosenjit, “Clustering-Based Recommender System Using Principles of Voting Theory”, Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014. 10.1109/IC3I.2014.7019655, 2014.
14. M. Rege, M. Dong, and F. Fotouhi, ‘‘Co-clustering documents and words using bipartite isoperimetric graph partitioning,’’ in Proc. Int. Conf. Data Mining pp. 532–541, 2006.
15. Y. Chen, M. Dong, and W. Wan, ‘‘Image co-clustering with multi-modality features and user feedbacks,’’ in Proc. Int. Conf. Multimedia, pp. 689–692, 2009.
16. P. F. Felzenszwalb and D. P. Huttenlocher, ‘‘Efficient graph-based image segmentation,’’ Int. J. Comput. Vis., vol. 59, no. 2, pp. 167–181, Sep. 2004.
17. J. Luo, B. Liu, B. Cao, and S. Wang, ‘‘Identifying miRNA-mRNA regulatory modules based on overlapping neighborhood expansion from multiple types of genomic data,’’ in Proc. Int. Conf. Intell. Comput, pp. 234–246, 2016.
18. G. Pio, M. Ceci, C. Loglisci, D. D’Elia, and D. Malerba, ‘‘Hierarchical and overlapping co-clustering of mRNA: miRNA interactions,’’ in Proc. Eur. Conf. Artif. Intell, pp. 654–659, 2012.
19. Hadrienj.github.io, “Deep Learning Book Series · 2.8 Singular Value Decomposition”, Github, Nov. 2019, https://hadrienj.github.io/assets/images/2.8/singular-value-decomposition.png
20. Andre Luiz Vizine Pereira, Eduardo Raul Hruschka, “Simultaneous co-clustering and learning to address the cold start problem in recommender systems”, Knowledge-Based Systems, Volume 82, Pages 11-19, ISSN 0950-7051, <https://doi.org/10.1016/j.knosys.2015.02.016>, 2015.
21. Hosseinzadeh Aghdam, Mehdi & Analoui, Morteza & Kabiri, Peyman. (2012). “Application of nonnegative matrix factorization in recommender systems”, 873-876. 10.1109/ISTEL.2012.6483108, 2012.
22. Shriya Gupta, “Top 5 Distance Similarity Measures implementation in Machine Learning”, Medium, Sep. 2019, https://miro.medium.com/max/875/1\*L1pWK9foGvUIT7uQM9f\_yQ.
23. “Euclidean Distance”, Wikipedia, 2021, https://upload.wikimedia.org/wikipedia/commons/thumb/5/55/Euclidean\_distance\_2d.svg/450px-Euclidean\_distance\_2d.svg.png
24. Shriya Gupta, “Top 5 Distance Similarity Measures implementation in Machine Learning”, Medium, Sep. 2019, [https://miro.medium.com/max/790/1\*dMv1HKYgFLlcCR5-ejLwPw.png](https://miro.medium.com/max/790/1*dMv1HKYgFLlcCR5-ejLwPw.png)
25. Vimarsh Karbhari, “What is a cosine similarity matrix?”, Medium, Jan. 2020, https://miro.medium.com/max/625/1\*dGWOzgAYv9NUkWvkETQUTQ.png
26. Tarlan Ahadli, “Naive Bayes Classifier: Bayesian Inference, Central Limit Theorem, Python/C++ Implementation”, Towards Data Science, Apr. 2020, https://miro.medium.com/max/875/1\*HfG1PY5-VSILokC66mLtsA.png
27. Namita Mutha, “Bernoulli Naive Bayes”, opengenus.org, <https://iq.opengenus.org/bernoulli-naive-bayes/#:~:text=Bernoulli%20Naive%20Bayes%20is%20used%20for%20discrete%20data,or%20failure%2C%200%20or%201%20and%20so%20on>.
28. Rohith Gandhi, “Naive Bayes Classifier”, Towards Data Science, May 2018, https://miro.medium.com/max/1576/1\*0If5Mey7FnW\_RktMM5BkaQ.png
29. Ajesh, A & Nair, Jayashree & Jijin, P, “A random forest approach for rating-based recommender system”. 1293-1297. 10.1109/ICACCI.2016.7732225, 2016.
30. Al-Molegi, Abdulrahman & Alsmadi, Izzat & Hassan, Najadat & Al-bashiri, Hael, “Automatic Learning of Arabic Text Categorization”, International Journal of Digital Contents and Applications. 2. 1-16. 10.21742/ijdcasd.2015.2.1.01, 2015.
31. Dr Stylianos (Stelios) Kampakis, “Performance measures: RMSE and MAE”, The Data Scientist, <https://thedatascientist.com/performance-measures-rmse-mae/>
32. JJ, “MAE and RMSE-Which Metric is Better?”, Medium, March 2016, [https://miro.medium.com/max/630/1\*OVlFLnMwHDx08PHzqlBDag.gif](https://miro.medium.com/max/630/1*OVlFLnMwHDx08PHzqlBDag.gif).
33. [Akhilendra](https://akhilendra.com/author/blogakhilendranewrole/), “Evaluation Metrics for Regression models- MAE Vs MSE Vs RMSE vs RMSLE” akhilendra.com, 2021, <https://akhilendra.com/evaluation-metrics-regression-mae-mse-rmse-rmsle/>
34. Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, Georg Lausen, “Book-Crossing Dataset”, Proceedings of the 14th International World Wide Web Conference (WWW '05), May 10-14, Chiba, Japan, http://www2.informatik.uni-freiburg.de/~cziegler/BX/, 2005.
35. Gipp, Bela & Beel, Joeran & Hentschel, Christian, “Scienstein: A Research Paper Recommender System”, 2009.
36. Tao Li, Jiandong Wang, Huiping Chen, Xinyu Feng and Feiyue Ye, "A NMF-based Collaborative Filtering Recommendation Algorithm", 6th World Congress on Intelligent Control and Automation, pp. 6082-6086, doi: 10.1109/WCICA.2006.1714249, 2006.
37. Sahu, Satya & Nautiyal, Anand & Prasad, Mahendra, “Machine Learning Algorithms for Recommender System - a comparative analysis”. International Journal of Computer Applications Technology and Research. 6. 97-100. 10.7753/IJCATR0602.1005, 2017.
38. Portugal, Ivens & Alencar, Paulo & Cowan, Donald, “The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review”. Expert Systems with Applications. 97.10.1016/j.eswa.2017.12.020, 2015.
39. Nguyen, Sang, “Model-Based Book Recommender Systems using Naïve Bayes enhanced with Optimal Feature Selection”. 217-222. 10.1145/3316615.3316727, 2019.
40. Gaudani, Hetal, “A Review Paper on Machine Learning Based Recommendation System. Development”. 2. 3955-3961, 2014.
41. Lampropoulos, Aristomenis & Tsihrintzis, George, “Review of Previous Work Related to Recommender Systems”. Intelligent Systems Reference Library. 92. 13-30. 10.1007/978-3-319-19135-5\_2, 2015.
42. A. Nawrocka, A. Kot and M. Nawrocki, "Application of machine learning in recommendation systems," 19th International Carpathian Control Conference (ICCC), 2018, pp. 328-331, doi: 10.1109/CarpathianCC.2018.8399650, 2018.
43. Mohammadreza Babaee, Stefanos Tsoukalas, Maryam Babaee, Gerhard Rigoll, Mihai Datcu, “Discriminative Nonnegative Matrix Factorization for dimensionality reduction”, Neurocomputing, Volume 173, Part 2, Pages 212-223, ISSN 0925-2312, <https://doi.org/10.1016/j.neucom.2014.12.124>, 2016.
44. Anwar, Khalid & Siddiqui, Jamshed & Sohail, Shahab, “Machine Learning Techniques for Book Recommendation: An Overview”, SSRN Electronic Journal. 10.2139/ssrn.3356349, 2019.
45. Joeran Beel, Bela Gipp, Stefan Langer & Corinna, “Research-paper recommender systems: a literature survey”, Int J Digit Libr 17, 305–338, <https://doi.org/10.1007/s00799-015-0156-0>, 2016.
46. Joy Lal Sarkar, Abhishek Majumder, “A new point-of-interest approach based on multi-itinerary recommendation engine, Expert Systems with Applications”, Volume 181, 115026, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2021.115026>, 2021
47. Al Fararni K., Aghoutane B., Riffi J., Sabri A., Yahyaouy A, “Comparative Study on Approaches of Recommendation Systems”. In: Bhateja V., Satapathy S., Satori H. (eds) Embedded Systems and Artificial Intelligence. Advances in Intelligent Systems and Computing, vol 1076. Springer, Singapore. https://doi.org/10.1007/978-981-15-0947-6\_72, 2020.
48. Piletskiy P., Chumachenko D., Meniailov I, “Development and Analysis of Intelligent Recommendation System Using Machine Learning Approach”. In: Nechyporuk M., Pavlikov V., Kritskiy D. (eds) Integrated Computer Technologies in Mechanical Engineering. Advances in Intelligent Systems and Computing, vol 1113. Springer, Cham. [https://doi.org/10.1007/978-3-030- 37618-5\_17](https://doi.org/10.1007/978-3-030-%2037618-5_17), 2020.
49. Javed, U., Shaukat, K., A. Hameed, I., Iqbal, F., Mahboob Alam, T. & Luo, S, “A Review of Content-Based and Context-Based Recommendation Systems”. International Journal of Emerging Technologies in Learning (iJET), 16(3), 274-306. Kassel, Germany: International Journal of Emerging Technology in Learning. Retrieved June 24, 2021 from https://www.learntechlib.org/p/219036/, 2021.