Taylor & Francis
Taylor & Francis Group

Automatika
Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

An approach to improve the accuracy of rating prediction for recommender systems

Thon-Da Nguyen

To cite this article: Thon-Da Nguyen (2024) An approach to improve the accuracy of rating prediction for recommender systems, Automatika, 65:1, 58-72, DOI: 10.1080/00051144.2023.2284026

To link to this article: https://doi.org/10.1080/00051144.2023.2284026

Published online: 21 Nov 2023.

AUTOMATIKA
2024, VOL. 65, NO. 1, 58–72
https://doi.org/10.1080/00051144.2023.2284026

An approach to improve the accuracy of rating prediction for recommender systems

Thon-Da Nguyen

University of Economics and Law, Ho Chi Minh City, Vietnam; Vietnam National University, Ho Chi Minh City, Vietnam

ABSTRACT
Sentiment analysis is critical for classifying users on social media and reviewing products through comments and reviews. At the same time, rating prediction is a popular and valuable topic in research on recommendation systems. This study improves the accuracy of ratings in recommendation systems through the combination of rating prediction and sentiment analysis from customer reviews. New ratings have been generated based on original ratings and sentiment analysis. Experimental results show that in almost all cases, revised ratings using a deep learning-based algorithm called LightGCN on 7 various real-life datasets improve rating prediction. In particular, rating prediction metrics (RMSE and MAE, R2, and explained variance) of the proposed approach (with revised ratings) are better than those of the typical approach (with unrevised ratings). Furthermore, evaluating ranking metrics (also top-k item recommendation metrics) for this model also shows that our proposed approach (with revised ratings) is much more effective than the original approach (with unrevised ratings). Our significant contribution to this research is to propose a better rating prediction model that uses a supplement factor sentiment score to enhance the accuracy of rating prediction.

ARTICLE HISTORY
Received 17 June 2022
Accepted 12 November 2023

KEYWORDS
VADER; sentiment analysis; recommender systems; rating prediction; recommendation systems

1. Introduction

The goal of recommender systems is to provide users with a list of recommendations for items. These systems are helpful in decision-making, maximizing profits for businesses, or minimizing risks. A recommendation system collects information from many essential data sources, such as (1) data based on content from products the user has purchased or used, (2) data from user profiles, (3) memory-based data collected from customer preferences, and (4) data provided by devices, e.g. sensors, location, and sale area. Recommendation system algorithms include content-based recommendation, collaborative filtering (CF), and hybrid approaches. CF is one of the most widely valuable techniques to predict an item, while content-based methods perform prediction based on the characteristics of the items chosen by customers [1]. The primary motivation of this research is that we want to find a better solution for rating prediction. We expect that our proposal can support users in accessing the right and suitable products on a website, primarily in e-commerce. There are a few reasons for this motivation. The first reason is that there has been an increasing number of e-commerce websites and their customers on the internet. Consumers tend to use products based on popular trends from other users. The second reason is that product recommender systems currently have an essential task.

This paper addresses the shortcomings of the conventional approach for rating prediction in recommendation systems. Our main contribution is introducing a new method that supports improved rating prediction. In particular, with the addition of the score subparameter of sentiment analysis, this study achieved good results from evaluating rating prediction metrics such as MAE, RMSE, RSquared, and Explained Variance (Exp Var).

After Section 1, Section 2 investigates relevant theories and knowledge. Next, Section 3 introduces related work. Then, in Section 4, we propose our approach for improving rating prediction. In Section 5, we present the analysis of the experimental results. Finally, Section 6 concludes with suggestions for future work.

2. Background

Rating prediction is an issue that is closely related to information filter approaches. The input data of this approach is the evaluation of users based on grading systems such as the start system. Then, the rating information is stored in a rating matrix to attempt to solve recommender systems based on rating prediction.

2.1. *Machine learning-based algorithms for rating prediction*

According to [2], rating prediction algorithms built into Surprise can be classified into the following five groups.

CONTACT Thon-Da Nguyen dant@uel.edu.vn

- **Group 1:** Basic algorithms (NormalPredictor, BaselineOnly)
- **Group 2:** k-NN inspired algorithms (KNNBasic, KNNWithMeans, KNNWithZScore, KNNBaseline)
- **Group 3:** Matrix factorization-based algorithms (SVD, SVDpp)
- **Group 4:** SlopeOne algorithm.
- **Group 5:** Algorithms based on coclustering.

Surprise is a state-of-the-art Python library for building and analysing rating prediction algorithms [2]. Currently, it is the only library that provides a scikit-learning-like API and uses metrics such as RMSE and MEA to evaluate the performance of the prediction.

2.2. Recent deep learning-based algorithms for rating prediction

FastAI Embedding Dot Bias (FAST)[1]: This is a model for recommender systems using embeddings and biases for users and items.

Neural Collaborative Filtering (NCF) [3]: This is a deep learning-based model with enhanced performance for user/item implicit feedback. It could ensemble generalized matrix factorization (GMF) and multilayer perceptron (MLP) to unify the strengths of linearity of MF and nonlinearity of MLP for modelling the user–item latent structures.

Multinomial VAE [4]: This is a generative model for recommender systems to predict user/item interactions.

Extreme Deep Factorization Machine (xDeepFM) [5]: This is a model that is based on deep learning for implicit and explicit feedback with user/item features.

Convolutional Sequence Embedding Recommendation (Caser) [6]: This model is based on convolutions. Its goal is to capture both the user's general preferences and sequential patterns.

Attentive Asynchronous Singular Value Decomposition (A2SVD) [7]: Sequential recommender systems take the sequence of user behaviours as context. This is a deep learning-based model that aims at capturing both long- and short-term user preferences for precise recommender systems. Its goal is to predict the items with which the user will interact within a short time.

GRU4Rec [7]: This is a sequential-based algorithm. Its goal is to capture both long- and short-term user preferences using recurrent neural networks.

Short-term and Long-term Preference Integrated Recommender (SLi-Rec) [7]: This is a sequential-based model. Its goal is to capture both long- and short-term user preferences using an attention mechanism, a time-aware controller and a content-aware controller.

Next Item Recommendation (NextItNet) [8]: This is a model based on dilated convolutions and a residual network that aims to capture sequential patterns. The model considers both user/item interactions and features.

Sequential Recommendation Via Personalized Transformer (SSEPT) [9]: This is a class of sequential recommendation that uses the transformer technique for encoding the user preference represented in terms of a sequence of items purchased/viewed before.

LightGCN [10]: This is a deep learning-based algorithm that simplifies the design of GCN for predicting implicit feedback. In this case, GCNs are networks that can learn patterns in graph information. GCNs are particularly well suited for recommendation systems due to their ability to encode relationships.

Multi-Interest-Aware Sequential User Modelling (SUM) [11]: This is an enhanced memory network-based sequential user model. Its purpose is to capture users' multiple interests.

Bilateral Variational Autoencoder (BiVAE) [12].

This is a generative model for dyadic data for collaborative filtering. The implementation of the model is also from Cornac, which is a framework for multimodal recommender systems focusing on models that utilize auxiliary data.

The Recommenders repository[2] provides examples and best practices for building recommendation systems using machine learning and deep learning-based models. The module recommenders contain functions to simplify common tasks that are executed in the development and evaluation of recommender systems.

2.3. Sentiment analysis

Sentiment analysis (known as opinion mining) is necessary in various fields, especially when studying customers, to gather opinions about companies' products and services [13]. In addition, it is also used in public areas such as national security and other public sectors.

For sentiment analysis, we use the library VADER (Valence Aware Dictionary for Enticement Reasoning) [14]. Recently, numerous works have utilized VADER as a tool for data analysis. There is cryptocurrency price prediction using tweet volumes by [15], analysis for the COVID-19 vaccine Tweets by [16], predictive analysis of resource usage data in academic libraries by [17], acceptance decision prediction in peer review by [18], predictions of customer response sentiment by [19], and student evaluations of teaching by [20]. Therefore, VADER would be a good choice for intensively researching and discovering its strengths.

According to [14], scores are categorized into fundamental types.

The sentiment is positive if the compound score is more than or equal to 0.05.

The sentiment is neutral if (the compound score is more than -0.05) and (the compound score is less than 0.05).

Sample Review 1:
"I had some issues with my 18-watt all tube amp so I experimented and replaced all the tubes. Issues not only continued but worsened. Hum, noise, volume fade, unwanted distortion at low volumes. I eventually put back the original El84s and that didn't help. Once I removed the Tung-Sol 12AX7 preamp tubes and replaced, the hum stopped, sound quality of the amp was consistent, quiet and back to par. My experience with the Tung-Sol tubes was very, very negative. They caused me to go out and buy another amp. I had no idea that brand new tubes could sound so badly that fast. I'll never buy this brand again."
{'neg': 0.216, 'neu': 0.762, 'pos': 0.022, 'compound': -0.9784}

Sample Review 2:
"These picks seem well made and have a solid feel, but I'm not sure I like the sound of them. Like so many other guitarists, I go through a lot of picks, searching for what feels and sounds the best for what I do. I don't this is it for me--at least not on acoustic. Whatever I didn't like about them got drowned out when I tried electric. Since I've been playing more acoustic lately, I might or might not buy these again."
{'neg': 0.13, 'neu': 0.731, 'pos': 0.139, 'compound': -0.0059}

Sample Review 3:
Great price but very fragile. I'm a 30 year radio pro always careful about my cables. The end separated from the cable - not very good quality, would not buy this brand again.
{'neg': 0.091, 'neu': 0.792, 'pos': 0.117, 'compound': 0.0041}

Figure 1. Two reviews with negative compound scores and one review with close to zero compound score.

Sample Review 4:
"Light weight, compact, and all in one tool, what more could I ask for?This tool does three things very well, turns tuners on stringed instruments quickly, cuts musical instrument strings, and helps pull acoustic guitar string pins. What's not to like? Well the cutter part of the tool is a little hard to open - guess I'm used to side cutters or pliers - and it doesn't cut strings really close to the tuning pin - it leaves about a half inch of string hanging. Oh, if you actually follow the three loops of string theory for stringing your guitar (go to the Elixir Strings website and watch the excellent guitar stringing videos from the Taylor guitar factory) your wrist will get a little tired turning the crank for the tuners. The alternative is a battery operated turner that weighs a ton more.I find it very hard to believe that with normal use on stringed instrument strings, somebody could manage to break this plastic tool. The tool is made of sturdy plastic. If abused, trying to cut copper wire, it might break; but the tool was never designed for that use.I've used this tool now for several months and expect that it will last for a very long time. I use this on three different guitars, an acoustic and two electrics (Les Paul and PRS). The pin puller works just fine, I actually put a rag on the guitar body to prevent any dents in the wood. The tool fits inside my hard shell Gibson guitar case. This is a very nice tool."
{'neg': 0.052, 'neu': 0.897, 'pos': 0.051, 'compound': 0.5002}

Figure 2. One sample review with a compound score of approximately 0.5.

Sample Review 5:
"Like another reviewer here, I too own a Fender G-Dec 30 (BTW: now going up for sale). The Spider Jam 75W specs are all out there so I won't repeat all that info but give you my users impressions instead. First I was fortunate enough to buy one for $300. (the same price as the Fender G-Dec 30), so it's an even simpler comparison for me.***The Line 6 Spider Jam 75W is a full head & shoulders taller and better than the G-Dec.***As just simply an amp it has more power & as good of sound. As for the rest, I'll take line 6's real drums sets and myriad of song backings over the Fender's midi/synthe tunes any day. The ability to save your custom tunes as JAM or WAV files internally or out to a larger SD card is a godsend and to me, the single biggest factor to own this amp over a G-Dec. (I use the max 2gb SD - note: it must be fat 16 formatted & is not HC.) The overdub layering is simply superb (though it's saved as mono). The XLR mike, 1/4" plug aux & MP3 inputs allow a great flexibility for total song creation. Coupled with my Line 6 Variax guitar, I have been able to produce an amazing range of sounds and songs.The only criticisms I can muster are: 1) - The menu process is more difficult than the G-Dec but easy enough with hands-on experience. 2) - The chrome plastic knobs look somewhat cheap but work well enough. 3) - Line six really needs to update to an SDHC format so it can utilize larger than 2gb SD storage cards. I have experimented with using a 4GB non-HC SD card & though I was able to format it, I was not able to read / write to it. Line 6, please go to SDHC! I would also like to see an improvement to an "incremented" or notched amp model selector pot. As every once in a while it will jump from the A or B selection if not exact."
{'neg': 0.015, 'neu': 0.804, 'pos': 0.181, 'compound': 0.997}

Figure 3. One sample review with a compound score higher than 0.99.

The sentiment is negative if the compound score is less than or equal to -0.05.

The value of the compound score is defined by the range from -1 (completely negative meaning extremely negative or not entirely supportive) to $+1$ (utterly extremely positive, fully supportive). The positive, neutral, and negative scores are the ratios for the percentage of text in each category. For example, different points of view can include positive or negative ideas in different proportions of text that are neutral between positive and negative states. A few typical examples of sentiment analysis are illustrated in Figures 1–3.

2.4. Evaluation metrics

The two kinds of evaluation metrics applied for this study are ranking metrics (including precision@k, recall@k, normalized discounted cumulative gain@k

(NDCG@k), and mean-average-precision (MAP)) and rating metrics (including root mean squared error (RMSE), mean average error (MAE), R squared, and explained variance). The details that are relevant to these evaluation metrics are described below.

2.5. RMSE and MAE

RMSE and MAE are suitable for observed ratings. This also means that both of these metrics are helpful for rating prediction. They are computed from Equations (1) and (2), respectively.

$$RMSE = \sqrt{\frac{1}{|\hat{R}|}\sum_{\hat{r}_{up}\in\hat{R}}(r_{up}-\hat{r}_{up})^2} \quad (1)$$

$$MAE = \frac{1}{|\hat{R}|}\sum_{\hat{r}_{up}\in\hat{R}}|r_{up}-\hat{r}_{up}| \quad (2)$$

where r_{up} is the true rating of user u for product p; $\hat{r}_{up}$ is the predicted rating of user u for product p. Meanwhile, $\hat{R}$ is the set of predicted ratings.

2.6. R squared (known as R2)

The R-squared (also known as R2) shows the ratio of the variance for a dependent variable used in a regression model. This metric explains the extent to which the variance of one variable explains the variance of the second variable.

The formula for R-squared is shown as follows.

$$R^2 = \frac{Unexplained\ Variantion}{Total\ Variation} \quad (3)$$

where

Unexplained Variation: Calculating predicted values, subtracting actual values, and squaring the results yields a list of squared errors, which is then summed and equals the unexplained variance.

Total variation: The average actual value and each of the actual values are subtracted, and then the results are squared and summed.

2.7. Explained variance

Explained variance is the proportion of explained variance and target variance. Explained variance is the subtraction of target variance and variance of prediction error. The proportion explained variance is defined by averaging the numbers; then, for each number, the mean is subtracted, and the results are squared. Then, the squares are found.

$$Explained\ Variance = \frac{Sum\ of\ squares\ between\ groups}{sum\ of\ squares\ total} \quad (4)$$

2.8. Precision@k

Precision@k is the ratio of recommended items in the top-k set that are relevant. The formula for Precision@k is defined as follows:

$$Precision@k = \frac{Number\ of\ recommended\ items@k\ that\ are\ relevant}{Number\ of\ recommended\ items\ @k} \quad (5)$$

2.9. Recall@k

Recall@k is the ratio of relevant items found in the top-k recommendations. The formula for Recall@k is defined as follows:

$$Recall@k = \frac{Number\ of\ recommend\ items@k\ that\ are\ relevant}{Total\ of\ relevant\ items} \quad (6)$$

2.10. Map@k

The mean average precision is defined from the average precision (AP). Map@k is calculated as follows: Step 1: Calculate the AP at an arbitrary threshold k of each dataset; Step 2: Sum up and find the mean of AP@k of every dataset to obtain mAP@k.

$$AP@k\frac{1}{N(k)}\sum_{i=1}^{k}\frac{TPseen(i)}{i} \quad (7)$$

where:

TP: true positives,

N(k) and TP can be computed as follows.

$$N(k) = min(k, TP_{total})$$

$$TPseen(i) = 0\ if\ i^{th} = False;$$

$$TPseen(i) = TP\ seen\ till\ i\ if\ i^{th} = True;$$

To compute mAP@k, we need to compute the average of the overall AP@k.

$$mAP@k = \frac{1}{N}\sum_{i=1}^{N}AP@k_i \quad (8)$$

2.11. nDGC@k

The normalized discounted cumulative gain (NDCG) is the DCG with a normalization factor in the denominator. The denominator is the ideal DCG score when recommending the most relevant items first.

$$NDCG@k = \frac{DCG@k}{IDCG@k} \quad (9)$$

in which

$$IDCG@k = \sum_{i=1}^{k^{ideal}} \frac{G_i^{ideal}}{log_2(i+1)} \tag{10}$$

In this paper, we use these kinds of metrics to evaluate the accuracy of our proposed approach compared with the traditional approach through machine learning algorithms for rating prediction and deep learning algorithms for recommending relevant items. This task is conducted on five real-life datasets and is described in later sections.

3. Related work

Currently, there are many exciting approaches for rating prediction, such as models based on convolutional neural networks (CNNs), latent Dirichlet allocation (LDA), long short-term memory (LSTM), similarities, graphs, sentiment analysis and even hybrid models.

In regard to rating prediction based on CNNs, [21–23] CNN models are used for rating prediction. [24] propose a rating prediction by considering four constituents regarding social networks: (1) user personal preferences, (2) interpersonal preference similarity, (3) mutual rating behaviour similarity, and (4) mutual rating behaviour spreading. These elements are merged to enhance the accuracy rating predictions. Their experimental results of our model show significant improvement on 7 real-life datasets. [25] use LSTM, which is an effective deep learning-based method, for sentiment analysis. Many researchers apply graph-based models for predicting ratings [26–29] and present rating predictions based on graphs. The authors [30] presented a tensor-based method to represent the relationship among reviewers, products and text features. Their experimental results showed that it is better to model reviewer and product information in the text-based learner. Their experimental results showed that their approach had significantly improved compared to several conventional methods, particularly for reviews with unusual items and inactive reviewers. [31] introduced an algorithm with the goal of enhancing rating prediction accuracy on seven real-life datasets (five Amazon datasets and two MovieLens datasets). Their experimental results show that their approach has obtained inevitable success. [32] introduced the model MJST based on LDA to analyse sentiment for sentiment analysis in microblogging. Nevertheless, the abovementioned traditional rating prediction methods often do not address review text, which is a vital channel for understanding and attracting users. [33] introduced a survey on rating prediction using deep learning techniques. In that paper, several methods using rating and review-text introduced, including work, were presented in detail. [34] proposed a model known as RBLT that showed a few contributions. Their findings on multiple real-world datasets prove that their model is better than several standard methods in terms of rating prediction. [35–39] present models based on sentiment analysis to build their recommender systems. [30] proposed a model using semantic similarities between datasets. Their findings show that their approach provides a significant performance improvement. [39] proposed a model that uses user sentiment (using LDA and the word2vec model), user topic similarity, and interpersonal influence for rating prediction. The authors of the paper [40] proposed a model that combines reviews and ratings. Their findings on many real-world datasets indicate that their model is better than several other methods.

Most abovementioned models utilized state-of-the-art approaches for rating prediction on recommender systems. [41–47] proposed models for analysing sentiment scores on Amazon review datasets. Recent deep learning-based algorithms for recommending relevant items include LightGCN [10], NCF, and Bivae. The experimental results on Microsoft Recommenders proved that LightGCN performs better than models SAR and NCF [10] when evaluated through metrics MAP@10, nDCG@10, Precision@10, and Recall@10. More specifically, running with approximately 1 million instances, LightGCN outperforms both algorithms (SAR, NCF) in terms of both accuracy and recommendation time performance, taking approximately 1 s compared with approximately 3 s and approximately 85 s for SAR and NCF, respectively. In addition, the experimental results on Microsoft Recommenders proved that Bivae is the best algorithm compared to LightGCN, SAR and NCF when running the algorithms on the MovieLens (100 K) dataset for 15 epochs. However, it is better to add more information from rating prediction. Our proposal differs from the abovementioned approaches mainly because we have used a supplementing factor that supports traditional rating prediction. Regarding the research gap, many papers only focus on issues relating to either rating prediction or sentiment analysis without paying attention to combining these two approaches. Our proposal in this paper aims to address that problem.

4. Methodology

The proposed approach of this study can be divided into 5 steps, as shown in Figure 4. Concise steps are described below.

Step 1: Collecting data

Seven real datasets are used for this study and are obtained from [48].

*Dataset ***Epinions*** comes from [49][3]: (includes 50,000 instances in this study). Features of the dataset used for this study are described as follows.

userId is the ID of the user, e.g. *chris_baehr*

itemId is the ID of a product, e.g. *Minolta_QMS_Page Pro_1250E_Printers*

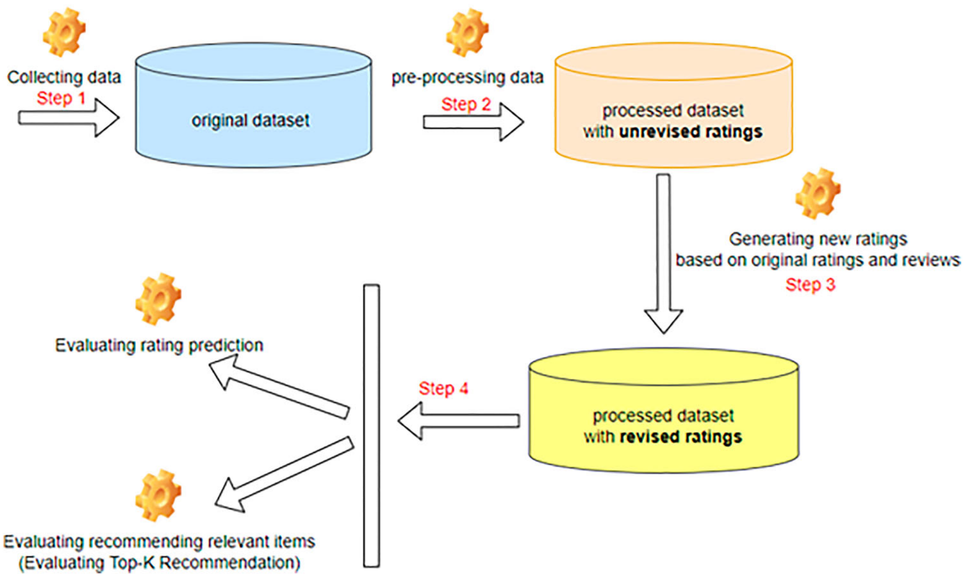

Figure 4. A proposed approach for improving rating prediction in recommender systems.

review ID is the text of the review which were given by users, e.g. *this is my first laptop and I bought it about two months ago as a portable desktop replacement to go with me when I travel I got it at best buy on sale for right at 1000, and it was the best deal on the market as far as I could tell.*
stars is the rating of a book on which users voted, e.g. *3.0.*

In the scope of this study, the author used the first 20 K instances of the *Epinions dataset* for rating prediction and ranking the top-K recommender.

*Dataset ***Good Read Reviews***[4]: (includes 50,000 instances in this study). Features of the dataset used for this study are described as follows.

user_id is the ID of the user, e.g. 2171432f539919a8e5f5 039a37b1837d.
book_id is the ID of a book, e.g. 34508.
review_text is the text of the review that was given by users, e.g. "*Good story… like Cohen… very amusing… Rincewind… sometimes find hard to like*".
rating is the rating of a book on which users voted, e.g. *5.0.*

In the scope of this study, the author used the 50 K first instances of the *Good Read Reviews* dataset for rating prediction and ranking the top-K recommender.

*Dataset ***Luxury Beauty***[5] (includes 34,278 instances). Features of the dataset used for this study are described as follows.

reviewerID: ID of the reviewer, e.g. A2HOI48JK8838M
asin - ID of the Luxury Beauty product, e.g. B00004U-9V2
Overall: rating of the Luxury Beauty product, e.g. 3.0
reviewText: text of the review, e.g. "There is no evidence to me that this product is an improvement over many others that are similarly priced… or less."

*Dataset ***Amazon Instant video***[6] (includes 37,126 instances). Features of the dataset used for this study are described as follows.

reviewerID: ID of the reviewer, e.g. A3NFIJUVEAJGP
asin - ID of the Luxury Beauty product, e.g. B000H4Y-NM0
Overall: rating of the Luxury Beauty product, e.g. 1.0
reviewText: text of the review, e.g. "maybe I just don't get it… but crude not funny and irritating most of the time… .guys that dumb shouldn't be on TV."

*Dataset ***Office Products***[7] (includes 53,258 instances). Features of the dataset used for this study are described as follows.

reviewerID: ID of the reviewer, e.g. A2PATWWZAX-HQYA
asin - ID of the office product, e.g. B000I0VMMC
Overall: rating of the office product, e.g. 4.0
reviewText: text of the review, e.g. "This is a great set for highlighting notes. Naturally, I like the yellow but the other colours work very well. The red and orange are very bright tones and bring the section marked immediately to attention. The retraction of the sharpie makes it easier to use (than having to put the cap back on and off). Very useful set - and I have the feeling I might just be fine for now… ..a great value for the set - and a good to have for your home office!"

*Dataset ***Digital Music***[8] (includes 64,706 instances). Features of the dataset used for this study are described as follows.

reviewerID: ID of the reviewer, e.g. AWHMBKCAMA-8KG
asin - ID of the digital music product, e.g. B000001Y15
Overall: rating of the digital music product, e.g. 5.0
reviewText: text of the review, e.g. "Makaveli's (2 Pac's) 7 Day Theory is one of 2 Pac's greatest albums.

Including hits such as Hail Mary or To Live And Die In L.A. this is a must have classic."
*Dataset ***Industrial and Scientific***[9] (includes 77,071 instances)

Features of the dataset used for this study are described as follows.

reviewerID: ID of the reviewer, e.g. ADQ073QJ0E5TK
asin - ID of the XXXX product, e.g. B00004RHKX
Overall: rating of the XXX product, e.g. 2.0
reviewText: text of the review, e.g. "Having purchased an already expensive shop-vac that came with a tiny 1/2″ or 1″ hose I was already disgruntled about buying this. It works the way it should, but my shop-vac has lost suction due to the increase in hose diameter, and it doesn't work as great as it should. Needless to say I will be purchasing a Craftsman shop-vac down the road.".

Step 2. Preprocessing data
Data split: The data are split into training and test sets. The split ratios are 75–25 for the training and testing datasets. The splitting is stratified based on items.

With all these datasets, all duplicated instances will be removed. The main features will be changed into a standard form to be convenient for comparison. Standardized features include *userID*, *itemID*, *Rating*, and *reviewText*, where *userID* is the ID of a user, *itemID* is the ID of an item, Rating is the rating of user *userID* towards item *itemID*, and *reviewText* is a review of user userID towards item *itemID*.

Step 3. Generating new ratings based on original ratings and reviews
The author of this study has built the following procedure to generate revised ratings by adding a factor that is based on sentiment analysis.

Procedure Calculating_Revised_Rating
Input: The input data of the procedure are a recommendation dataset that contains a feature called *Rating0* (unrevised ratings, which were voted by users) and another feature called *reviewText* (reviews that were reviewed by users). In addition, another input value is *beta_coeff*, which has values of 0.5, 1.0, and 1.5.
Output: A new recommendation dataset similar to the original when inputting one with new features called Rating1, Rating2 and Rating3 (revised ratings)

The process of the procedure, **Procedure Calculating_Revised_Rating**, is illustrated as follows.

```
BEGIN Procedure Calculating_Revised_Rating
feature['Sentiment_Score']←Calculation_Sentiment(feature['reviewText']
# Generating the feature Rating1
beta_coeff ←0.5
feature['Rating1']←feature['Rating0'] + beta_coeff∗feature['Sentiment_
   Score']
IF feature['Rating1'] < 1 THEN
   feature['Rating1']←1
ELSE IF feature['Rating1'] > 5 THEN
   feature['Rating1']←5
# Generating c the feature Rating2
beta_coeff ←1
feature['Rating2']←feature['Rating0'] + beta_coeff∗feature['Sentiment_
   Score']
IF feature['Rating2'] < 1 THEN
   feature['Rating2']←1
ELSE IF feature['Rating2'] > 5 THEN
   feature['Rating2']←5
# Generating the feature Rating3
beta_coeff ←1.5
feature['Rating3']←feature['Rating0'] + beta_coeff∗feature['Sentiment_
   Score']
IF feature['Rating3'] < 1 THEN
   feature['Rating3']←1
ELSE IF feature['Rating3'] > 5 THEN
   feature['Rating3']←5
Show_result(RDB)
END Procedure Calculating_Revised_Rating
```

The function Calculation_Sentiment is defined by Vader sentiment analysis. It returns the value of −1, 0, and 1 if the review is negative, neutral, and positive, respectively. The procedure Show_result shows all instances with all features, such as userID, itemID, Rating1, Rating2, and Rating3, which are generated from Rating0 through Vader sentiment analysis. In particular, Rating1's values would be increased 50% (when reviews were positive) or increased −50% of Rating0's values (when reviews were negative) or still would be unchanged Rating0's values when reviews were neutral. Similarly, Rating2's values would be increased 100% (when reviews were positive) or increased −100% of Rating0's values (when reviews were negative) or would remain unchanged Rating0's values when reviews were neutral. It is the same for Rating3. Note that if the values of Rating1, Rating2 or Ratung3 are less than 1 or more than 5, they are invalid values for rating (only obtain values of from 1 to 5).

Step 4. Evaluate rating prediction and recommending relevant items
The author of this study uses the techniques of Microsoft Recommender Systems[10] along with [2] to evaluate our proposed approach compared to the traditional approach. The evaluation is divided into two parts: one for evaluating rating prediction and the other for evaluating recommending relevant items. They are described below.

**Part 1: Evaluate rating prediction-based machine learning techniques*

In this evaluation, evaluation metrics such as RMSE, MAE, R2 (R Squared), and Explained Variance are used to compare two approaches: the traditional approach with the abovementioned datasets using unrevised ratings and our proposed approach with the abovementioned datasets using revised ratings. Regarding techniques, the state-of-the-art algorithm for rating prediction called SVDpp has been applied. This is also the best algorithm in the benchmark generated by Surpise, a Python scikit for recommender systems.

**Part 2: Evaluate recommending relevant items based on deep learning techniques*

In this part, 4 deep learning-based algorithms, LightGCN, Bivae, NCF, and SAR, have been used along with top-K recommendation metrics (MAP, nDGC@k, Precision@k, Recall@k) to evaluate recommending relevant items. This evaluation is also run on the 5 abovementioned datasets to compare two approaches such as those in Part 1.

5. Experimental results and analysis

The statistical results in terms of rating prediction obtained after executing the programme in Python language on the Google Colab Pro + environment are detailed in the following 17 tables along with RMSE, MAE, S Squared and Explained Variance metrics (for rating prediction) and top-k item recommendation metrics such as MAP@k, NDCG@k, Precision@k, Recall@k.

The figures in Table 1 show that when the SVDpp algorithm is applied on part of the Epinions dataset, the rating prediction metrics of Rating1 are better than those of Rating0 ($RMSE = 1.026431, MAE = 0.846389$, $R2 = 0.081207$ and $Exp\_var = 0.159369$ *in* comparison with $RMSE = 1.086762$, $MAE = 0.893165$, $R2 = 0.069371$ *and* $Exp\_var = 0.133033$, respectively). In addition, all four top-10 item recommendation metrics (Map@10, Ndcg@10, Precision@10, Recall@10) of all cases with revised ratings are also better than those of Rating0.

The figures in Table 2 show that when the SAR algorithm is applied on part of dataset Epinions, all cases with revised ratings are also better than those of Rating0 in terms of top-10 item recommendation metrics (Map@10, Ndcg@10, Precision@10, Recall@10).

Look at Table 3 and Table 4, it can be seen that when algorithms BIVAE and NCF are applied on part of dataset Epinions, all top-10 item recommendation metrics (Map@10, Ndcg@10, Precision@10, Recall@10) are similar in all cases (unrevised and revised ratings).

In Table 5, cases of revised ratings (Rating1, Rating2, Rating3) are better than the case of unrevised ratings (Rating0) in terms of rating prediction metrics such as RMSE and MAE. There is one case *(Rating3* with $beta\_coeff = 1.5$*)* where all four rating prediction metrics (RMSE, MAE, R2, Exp_Var) are better than that of the case of unrevised ratings (Rating0 with $beta\_coeff = 0$*)* ($RMSE = 2.273142$, $MAE = 1.75119$, $R2 = -2.201175$ and $Exp\_Var = -2.181318$ *in comparison with* $RMSE = 2.806461$, $MAE = 1.837605$, $R2 = -3.626541$ *and* $Exp\_Var = -3.103294$, *respectively).*

Table 1. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the SVDpp algorithm and the Epinions dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	RMSE	1.086762	1.026431	1.07707	1.087416
	MAE	0.893165	0.846389	0.889456	0.904573
	RSQUARED(R2)	0.069371	0.081207	0.049319	0.047375
	EXP_VAR	0.133033	0.159369	0.128928	0.117409
Item Recommendation Metrics (k = 10)	MAP@k	0.003754	0.010452	0.013407	0.01153
	NDCG@k	0.006079	0.013162	0.016027	0.015693
	Precision@k	0.00137	0.002283	0.002511	0.002968
	Recall@k	0.012557	0.021689	0.023973	0.028539

Table 2. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the SAR algorithm and the Epinions dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	3.089599	3.271399	3.491794	3.524586
	MAE	2.923187	3.137402	3.354052	3.373613
	RSQUARED(R2)	−8.164814	−11.13556	−11.868207	−11.02305
	EXP_VAR	0.02641	0.010568	−0.01068	−0.023051
Item Recommendation Metrics (k = 10)	**MAP@k**	0.002626	**0.003015**	**0.002975**	**0.002963**
	NDCG@k	0.003901	**0.004294**	**0.004251**	**0.004196**
	Precision@k	0.000911	**0.000958**	**0.000958**	**0.000911**
	Recall@k	0.007467	**0.007946**	**0.007946**	**0.007826**

Table 3. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the BIVAE algorithm and the Epinions dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	4.174634	4.271054	4.392025	4.413836
	MAE	4.000574	4.116412	4.232249	4.253846
	RSQUARED(R2)	−11.249013	−13.06672	−13.001964	−13.05186
	EXP_VAR	−0.000142	−0.000191	−0.000213	−0.000221
Item Recommendation Metrics (k = 10)	**MAP@k**	0.00248	**0.00248**	**0.00248**	**0.00248**
	NDCG@k	0.003773	**0.003773**	**0.003773**	**0.003773**
	Precision@k	0.000857	**0.000857**	**0.000857**	**0.000857**
	Recall@k	0.008031	**0.008031**	**0.008031**	**0.008031**

Table 4. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the NCF algorithm and the Epinions dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	4.099901	4.201992	4.325302	4.325736
	MAE	3.951241	4.063082	4.174924	4.174924
	RSQUARED(R2)	−14.069047	−15.63505	−14.770402	−14.68657
	EXP_VAR	−0.072983	−0.081669	−0.077518	−0.07472
Item Recommendation Metrics (k = 10)	**MAP@k**	**0.013514**	**0.013514**	**0.013514**	**0.013514**
	NDCG@k	**0.017052**	**0.017052**	**0.017052**	**0.017052**
	Precision@k	**0.002703**	**0.002703**	**0.002703**	**0.002703**
	Recall@k	**0.027027**	**0.027027**	**0.027027**	**0.027027**

Table 5. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the LightGCN algorithm and the Epinions dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	2.806461	**2.203601**	**2.593131**	**2.273142**
	MAE	1.837605	**1.516708**	**1.836426**	**1.75119**
	RSQUARED (R2)	−3.626541	−4.210842	−5.935665	**−2.201175**
	EXP_VAR	−3.103294	−4.207257	−4.330517	**−2.181318**
Item Recommendation Metrics (k = 10)	**MAP@k**	0.004838	0.003474	0.00442	0.004213
	NDCG@k	0.006278	0.0048	0.005391	0.005725
	Precision@k	0.001198	0.000958	0.00091	0.00115
	Recall@k	0.009982	0.009104	0.007507	0.010182

Table 6. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the SVDpp algorithm and the Good Read Reviews dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	0.923721	1.044644	1.041416	1.112978
	MAE	0.739926	0.791083	0.789602	0.820425
	RSQUARED(R2)	0.186135	0.159183	0.164371	0.133339
	EXP_VAR	0.186892	0.161029	0.166232	0.135305
Item Recommendation Metrics (k = 10)	**MAP@k**	0.003484	**0.004533**	0.000801	0.000998
	NDCG@k	0.007884	**0.008142**	0.002928	0.002371
	Precision@k	0.004545	0.003031	0.001818	0.001212
	Recall@k	0.009091	0.008667	0.003036	0.002768

Table 7. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the SAR algorithm and the Good Read Reviews dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	2.878051	3.024346	3.125247	3.153264
	MAE	2.699178	2.86881	2.921316	2.902238
	RSQUARED(R2)	−6.04335	−8.322579	**−5.704291**	**−4.710287**
	EXP_VAR	0.116831	0.065773	0.05595	0.03798
Item Recommendation Metrics (k = 10)	**MAP@k**	0.009151	0.008933	0.008868	0.008589
	NDCG@k	0.017025	**0.017558**	**0.017426**	0.016686
	Precision@k	0.007683	**0.008564**	**0.008942**	0.008438
	Recall@k	0.012712	**0.012883**	**0.013093**	0.012357

Considering Table 6, the item top-k recommendation metrics Map@10 and nDCG@10 of Rating1 (revised ratings with *beta_coeff* = *0.5*) are better than those of Rating0 (unrevised ratings with *beta_coeff* = *0*).

Information from Table 7 indicates that the rating prediction metric R2 of Rating2 and Rating3 (revised ratings with *beta_coeff* = *1 and beta_coeff* = *1.5*) is that of Rating0 (unrevised ratings with *beta_coeff* = *0*) *(R2* = *−5.704291* and R2 = *−5.704291* in comparison with *R2* = *−6.04335, respectively).* In addition, the top-k item recommendation metrics of unrevised ratings cases such as Rating1 and Rating2 are better than those of Rating0 (unrevised ratings).

Information from Table 8 and Table 9 shows that when the BIVAE and NCF algorithms are applied on part of the Good Read Reviews dataset, all top-10 item recommendation metrics (Map@10, Ndcg@10, Precision@10, Recall@10) are similar in all cases (unrevised and revised ratings). This is also the same as other datasets in this study when these have been used. In addition, the rating prediction metric R2 of Rating2 and Rating3 (revised ratings with *beta_coeff* = *1 and beta_coeff* = *1.5*) is that of Rating0 (unrevised ratings with *beta_coeff* = *0*) *(R2* = *−11.222369* and *R2* = *−10.05562* in comparison with *R2* = *−11.965421, respectively).*

Table 8. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the BIVAE algorithm and the Good Read Reviews dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics (k = 10)	**RMSE**	3.958388	4.119315	4.310465	4.363372
	MAE	3.802196	3.977012	4.151828	4.19458
	RSQUARED(R2)	−13.308904	−15.16414	−13.806997	−12.96849
	EXP_VAR	−0.106934	−0.097503	−0.069823	−0.059809
Item Recommendation Metrics (k = 10)	**MAP@k**	0.004111	**0.004111**	**0.004111**	**0.004111**
	NDCG@k	0.013205	**0.013205**	**0.013205**	**0.013205**
	Precision@k	0.009572	**0.009572**	**0.009572**	**0.009572**
	Recall@k	0.013257	**0.013257**	**0.013257**	**0.013257**

Table 9. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the NCF algorithm and the Good Read Reviews dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	3.927146	4.024574	4.152838	4.177163
	MAE	3.769898	3.872555	3.975212	3.978836
	RSQUARED(R2)	−11.965421	−12.79979	**−11.222369**	**−10.05562**
	EXP_VAR	−0.017517	−0.022818	−0.023194	−0.024897
Item Recommendation Metrics (k = 10)	**MAP@k**	0.014267	**0.014267**	**0.014267**	**0.014267**
	NDCG@k	0.024471	**0.024471**	**0.024471**	**0.024471**
	Precision@k	0.010084	**0.010084**	**0.010084**	**0.010084**
	Recall@k	0.03379	**0.03379**	**0.03379**	**0.03379**

Table 10. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the LightGCN algorithm and the Good Read Reviews dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	7.266877	**6.43516**	**6.43516**	**6.07993**
	MAE	5.73316	**5.170111**	**5.170111**	**4.642423**
	RSQUARED(R2)	−60.447655	**−43.21191**	**−43.211913**	**−28.92861**
	EXP_VAR	−25.456365	**−16.36399**	**−16.363986**	**−13.33891**
Item Recommendation Metrics (k = 10)	**MAP@k**	0.008632	**0.010949**	**0.010949**	**0.011917**
	NDCG@k	0.022053	**0.025216**	**0.025216**	**0.026423**
	Precision@k	0.012217	**0.01335**	**0.013854**	**0.013854**
	Recall@k	0.018952	**0.022447**	**0.025312**	**0.025312**

Table 11. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the SVDpp algorithm and the Luxury Beauty dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	0.50921	0.538391	0.586472	0.581163
	MAE	0.229382	0.206357	**0.213458**	**0.204202**
	RSQUARED(R2)	0.634959	0.632045	**0.754324**	**0.755728**
	EXP_VAR	0.635153	0.63205	**0.75433**	**0.755728**
Item Recommendation Metrics (k = 10)	**MAP@k**	0.002661	0.001766	**0.002663**	0.001985
	NDCG@k	0.003795	0.002966	**0.004857**	**0.00427**
	Precision@k	0.00094	0.000877	**0.001566**	**0.001535**
	Recall@k	0.00699	0.005951	**0.010635**	**0.010747**

Looking at Table 10, it can be seen that when the LightGCN algorithm is applied on part of the Good Read Reviews dataset, all rating prediction metrics and top-k item recommendation metrics of revised ratings Rating1, Rating2, and Rating3 are better than those of Rating0 (unrevised ratings). This is absolutely similar when the LightGCN algorithm is applied on the *Office Products* and *Digital Music* datasets.

Table 11 indicates that in the case of *Rating2*, almost all metrics are better when compared with *Rating0,* except the *RMSE* metric. Similarly, in the case of *Rating3*, almost all metrics are better when compared with Rating0, except the RMSE metric (for rating prediction) and the Map@k metric (for recommending)

Figures from Table 12 show that except for the Exp_Var metric, other rating prediction metrics of cases revised ratings (Rating1, Rating2 and Rating3) are better than that of Rating0. Moreover, in terms of top-k item recommendation metrics, the case Rating 2 (revised ratings with *beta_coeff = 0.5*) is better than the case Rating1 (unrevised ratings).

Table 13 shows that when the SVDpp algorithm is applied, the rating prediction metrics of case Rating2 are better than those of case Rating1 (*Map@k = 0.00985*, *nDCG@k = 0.015552*, *Precision@k = 0.004045* and *Recall@k = 0.02915* in comparison with *Map@k = 0.009414*, *nDCG@k = 0.014441*, *Precision@k = 0.00345* and *Recall@k = 0.027712, respectively).* Apart from

Table 12. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the LightGCN algorithm and the Luxury Beauty dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	3.096622	**2.970825**	**2.936649**	**2.930355**
	MAE	2.143879	**2.048836**	**2.010953**	**1.98236**
	RSQUARED(R2)	−6.608007	**−6.530148**	**−6.039463**	**−6.393526**
	EXP_VAR	−4.854175	−5.150562	−4.898777	−5.158885
Item Recommendation Metrics (k = 10)	**MAP@k**	0.162432	0.161652	**0.162535**	0.160931
	NDCG@k	0.184096	0.183366	**0.184622**	0.182422
	Precision@k	0.044302	0.044249	**0.044433**	0.044118
	Recall@k	0.217044	0.217117	**0.218721**	0.215084

Table 13. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the SVDpp algorithm and the Amazon Instant video dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	0.922629	1.142019	1.396575	1.399027
	MAE	0.682074	0.811094	0.968826	0.964397
	RSQUARED(R2)	0.209946	0.142226	0.096332	0.088224
	EXP_VAR	0.210425	0.142906	0.096972	0.08872
Item Recommendation Metrics (k = 10)	**MAP@k**	0.009414	**0.009883**	**0.00985**	0.009161
	NDCG@k	0.014441	**0.014662**	**0.015552**	0.013346
	Precision@k	0.00345	**0.00348**	**0.004045**	0.003153
	Recall@k	0.027712	0.026497	**0.02915**	0.022979

Table 14. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the SAR algorithm and the Amazon Instant video dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	2.845757	2.994129	3.052612	3.078504
	MAE	2.67478	2.81265	2.895872	2.929161
	RSQUARED(R2)	−6.18131	**−5.590325**	**−4.191345**	**−4.330128**
	EXP_VAR	−0.07783	−0.098103	**−0.076359**	−0.079651
Item Recommendation Metrics (k = 10)	**MAP@k**	0.151644	0.149308	0.147562	0.147749
	NDCG@k	0.205435	0.202281	0.19989	0.199941
	Precision@k	0.050799	0.04963	0.049142	0.049064
	Recall@k	0.304792	0.300654	0.297363	0.296908

Table 15. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the LightGCN algorithm and the Amazon Instant video dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	2.861483	**2.825397**	3.058275	3.024469
	MAE	2.170964	**2.10743**	2.273216	2.229723
	RSQUARED(R2)	−5.471047	**−4.037013**	**−3.411085**	**−3.433079**
	EXP_VAR	−2.590434	**−1.958619**	**−1.570642**	**−1.652674**
Item Recommendation Metrics (k = 10)	**MAP@k**	0.10279	**0.104477**	**0.10464**	**0.104252**
	NDCG@k	0.148198	**0.150833**	**0.150087**	**0.150102**
	Precision@k	0.040136	**0.040936**	0.040078	**0.040409**
	Recall@k	0.243817	**0.247493**	**0.245917**	**0.247007**

that, the case Rating1 is similar except for the Recall@k metric.

Table 14 shows that the rating prediction metric R2 of Rating1, Rating2 and Rating3 (revised ratings with *$beta\_coeff = 0.5$, $beta\_coeff = 1$ and $beta\_coeff = 1.5$*) is that of Rating0 (unrevised ratings with $beta\_coeff = 0$) *($R2 = -5.590325$, $R2 = -4.191345$ and $R2 = -4.330128$* in comparison with *$R2 = -6.18131$*, respectively*)*. This is the same as applying the SAR algorithms on datasets *Office Products*, *Digital Music*, *Industrial and Scientific*.

In Table 15, the rating prediction metrics and top-k item recommendation metrics of Rating1 are better than those of Rating0. Two rating prediction metrics (R and Exp_Var) of Rating1, Rating2 and Rating3 are better than those of Rating0. In addition, the top-k item prediction metrics of Rating3 are also better than those of Rating0 (*$Map@k = 0.104252$*, *$nDCG@k = 0.150102$*, *$Precision@k = 0.040409$* and *$Recall@k = 0.247007$* in comparison with *$Map@k = 0.10279$*, *$nDCG@k = 0.148198$*, *$Precision@k = 0.040136$* and *$Recall@k = 0.243817$*, *respectively)*.

The figures in Table 16 indicate that the top-k item prediction metrics of Rating 3 are also better than those of Rating0 (Map@k = 0.006135, nDCG@k = 0.009829, Precision@k = 0.002749 and Recall@k = 0.017954 in comparison with Map@k = 0.005592, nDCG@k = 0.008493, Precision@k =

Table 16. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the SVD algorithm and the Office Products dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	0.861829	0.983752	1.181223	1.1724
	MAE	0.638643	**0.618139**	0.709024	0.689271
	RSQUARED(R2)	0.121831	0.042718	0.01464	0.010454
	EXP_VAR	0.121832	0.042745	0.014649	0.010456
Item Recommendation Metrics (k = 10)	**MAP@k**	0.005592	0.004462	0.005163	**0.006135**
	NDCG@k	0.008493	0.007604	**0.008698**	**0.009829**
	Precision@k	0.002193	**0.002255**	**0.002719**	**0.002749**
	Recall@k	0.015302	0.014813	**0.016052**	**0.017954**

Table 17. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the SVDpp algorithm and the Industrial and Scientific dataset).

beta_coeff Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	0.859426	0.997283	1.187052	1.18108
	MAE	0.58371	0.649362	0.755215	0.743309
	RSQUARED(R2)	0.17927	0.122358	0.096661	0.09151
	EXP_VAR	0.179375	0.12246	0.096744	0.091603
Item Recommendation Metrics (k = 10)	**MAP@k**	0.000854	**0.001304**	**0.002128**	**0.001675**
	NDCG@k	0.001521	**0.002423**	**0.003545**	**0.0028**
	Precision@k	0.000507	**0.000806**	**0.000996**	**0.000833**
	Recall@k	0.002779	**0.004995**	**0.006968**	**0.005525**

Table 18. Comparison of evaluation metrics between the dataset with unrevised ratings and datasets with revised ratings (using the LightGCN algorithm and the *Industrial and Scientific dataset*).

Sentiment Coefficient (μ) Ratings		0 Rating0	0.5 Rating1	1 Rating2	1.5 Rating3
Rating Prediction Metrics	**RMSE**	4.330677	**4.289866**	4.529551	**4.28985**
	MAE	3.251464	**3.166583**	**3.324152**	**3.20465**
	RSQUARED(R2)	−22.442572	**−17.83001**	**−13.651156**	**−12.32775**
	EXP_VAR	−10.45332	**−8.652367**	**−6.581864**	**−5.717495**
Item Recommendation Metrics (k = 10)	**MAP@K**	0.051317	**0.053864**	**0.052724**	**0.051968**
	NDCG@K	0.06779	**0.070406**	**0.068825**	**0.06778**
	Precision@K	0.016422	**0.016685**	0.01625	0.016096
	Recall@K	0.096292	**0.098294**	0.095965	0.094253

0.002193 and Recall@k = 0.015302, respectively). Furthermore, the top-k item prediction metrics of Rating 3 (except for the Map@k metric) are also better than those of Rating0 (nDCG@k = 0.008698, Precision@k = 0.002719 and Recall@k = 0.016052 in comparison with nDCG@k = 0.008493, Precision@k = 0.002193 and Recall@k = 0.015302, respectively).

Considering Table 17, all top-k item prediction metrics of Rating1, Rating2, and Rating3 are better than those of Rating0.

Figures from Table 18 indicate that when the LightGCN algorithm is applied on part of the *Industrial and Scientific* dataset, all rating prediction metrics and top-k item recommendation metrics of Rating1 (revised ratings) are better than those of Rating0 (unrevised ratings).

Among the abovementioned machine learning-based algorithms and deep learning-based algorithms, the LightGCN algorithm is the best algorithm for both rating prediction and item recommendation when working with revised ratings. This algorithm could predict ratings effectively on the following real datasets:

(1) Dataset Epionions with Rating3 (beta_coff = 1.5)
(2) Dataset Good Read Reviews with Rating1 (beta_coff = 0.5), Rating2 (beta_coff = 1), and Rating3 (beta_coff = 1.5)
(3) Dataset Luxury Beauty with Rating1 (beta_coff = 0.5), Rating2 (beta_coff = 1), and Rating3
(4) (beta_coff = 1.5)
(5) Dataset Amazon Instant Videos with Rating1 (beta_coff = 0.5)
(6) Dataset Office Products with Rating1 (beta_coff = 0.5), Rating2 (beta_coff = 1), Rating3
(7) (beta_coff = 1.5)
(8) Dataset Digital Music with Rating1 (beta_coff = 0.5), Rating2 (beta_coff = 1), Rating3
(9) (beta_coff = 1.5)
(10) Dataset Industrial and Scientific with Rating1 (beta_coff = 0.5), Rating2 (beta_coff = 1), and Rating3 (beta_coff = 1.5)

In addition, the LightGCN algorithm could effectively recommend the top-k items on the following real datasets:

(1) Dataset Good Read Reviews with Rating1 (beta_coff = 0.5), Rating2 (beta_coff = 1), Rating3
(2) (beta_coff = 1.5)

(3) Dataset Luxury Beauty with Rating2 (beta_coff = 1.0)
(4) Dataset Amazon Instant Videos with Rating1 (beta_coff = 0.5), Rating3 (beta_coff = 1.5)
(5) Dataset Office Products with Rating1 (beta_coff = 0.5), Rating3 (beta_coff = 1.5)
(6) Dataset Digital Music with Rating1 (beta_coff = 0.5), Rating2 (beta_coff = 1), Rating3
(7) (beta_coff = 1.5)
(8) Dataset Industrial and Scientific with Rating1 (beta_coff = 0.5).

6. Conclusion

In this study, rating adjustment is applied through the addition of a review emotion analysis factor (if emotions are positive, the rating will be adjusted up, and vice versa, if the emotion is neutral, the value of the rating will still be unchanged). When adjusting the rating, one of the considered algorithms that has been used is the deep learning-based algorithm called LightGCN, which gave significant results for both rating prediction and top-k item recommendation. The algorithm could predict ratings better with revised ratings in 6 of the 7 real-life datasets mentioned above. Moreover, it could recommend the top-k items better on all 7 mentioned real-life datasets with revised ratings. In the future, we plan to improve our proposed approach by searching for novel solutions and applying our proposed approach to financial information systems, e-commerce and big data.

Notes

1. https://docs.fast.ai/collab.html
2. https://microsoft-recommenders.readthedocs.io/en/latest/
3. https://drive.google.com/file/d/1lxypFK_7gS0avjMQbzpfZe0IXW_mioUj/view
4. https://drive.google.com/uc?id=1pQnXa7DWLdeUpvUFsKusYzwbA5CAAZx7
5. https://jmcauley.ucsd.edu/data/amazon_v2/categoryFilesSmall/Luxury_Beauty_5.json.gz
6. http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Amazon_Instant_Video_5.json.gz
7. http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Office_Products_5.json.gz
8. http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Digital_Music_5.json.gz
9. https://jmcauley.ucsd.edu/data/amazon_v2/categoryFilesSmall/Industrial_and_Scientific_5.json.gz
10. https://github.com/microsoft/recommenders

Acknowledgement

This research is funded by the University of Economics and Law, VNU-HCM.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Thon-Da Nguyen http://orcid.org/0000-0002-2660-5011

References

[1] Kumar P, Thakur RS. Recommendation system techniques and related issues: a survey. Int J Inf Technol. 2018;10(4):495–501. doi:10.1007/s41870-018-0138-8
[2] Hug N. Surprise: a Python library for recommender systems. J Open Source Software. 2020;5(52):2174, doi:10.21105/joss.02174
[3] He X, Liao L, Zhang H, et al. Neural collaborative filtering. Paper presented at the Proceedings of the 26th international conference on world wide web; 2017.
[4] Liang D, Krishnan RG, Hoffman MD, et al. Variational autoencoders for collaborative filtering. Paper Presented at the Proceedings of the 2018 World Wide web Conference; 2018.
[5] Lian J, Zhou X, Zhang F, et al. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. Paper Presented at the Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2018.
[6] Tang J, Wang K. Personalized top-n sequential recommendation via convolutional sequence embedding. Paper Presented at the Proceedings of the Eleventh ACM International Conference on web Search and Data Mining; 2018.
[7] Yu Z, Lian J, Mahmoody A, et al. Adaptive user modeling with long and short-term preferences for personalized recommendation. Paper Presented at the IJCAI; 2019.
[8] Yuan F, Karatzoglou A, Arapakis I, et al. A simple convolutional generative network for next item recommendation. Paper Presented at the Proceedings of the Twelfth ACM International Conference on web Search and Data Mining; 2019.
[9] Wu L, Li S, Hsieh C-J, et al. SSE-PT: Sequential recommendation via personalized transformer. Paper Presented at the Proceedings of the 14th ACM Conference on Recommender Systems; 2020.
[10] He X, Deng K, Wang X, et al. Lightgcn: Simplifying and powering graph convolution network for recommendation. Paper Presented at the Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval; 2020.
[11] Lian J, Batal I, Liu Z, et al. (2021). Multi-Interest-Aware User Modeling for Large-Scale Sequential Recommendations. *arXiv preprint arXiv:2102.09211*.
[12] Truong Q-T, Salah A, Lauw HW. Bilateral variational autoencoder for collaborative filtering. Paper Presented at the Proceedings of the 14th ACM International Conference on Web Search and Data Mining; 2021.
[13] Yue L, Chen W, Li X, et al. A survey of sentiment analysis in social media. Knowl Inf Syst. 2019;60(2):617–663. doi:10.1007/s10115-018-1236-4
[14] Hutto C, Gilbert E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. Paper Presented at the Proceedings of the International AAAI Conference on Web and Social Media; 2014.
[15] Abraham J, Higdon D, Nelson J, et al. Cryptocurrency price prediction using tweet volumes and sentiment analysis. SMU Data Sci Rev. 2018;1(3):1.
[16] Sandaka GK, Gaekwade BN. (2021). Sentiment Analysis and Time-series Analysis for the COVID-19 vaccine Tweets.

[17] Deo GS, Mishra A, Jalaluddin ZM, et al. Predictive analysis of resource usage data in academic libraries using the VADER sentiment algorithm. Paper Presented at the 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN); 2020.
[18] Ribeiro AC, Sizo A, Lopes Cardoso H, et al. Acceptance decision prediction in peer-review through sentiment analysis. Paper Presented at the EPIA Conference on Artificial Intelligence; 2021.
[19] Borg A, Boldt M. Using VADER sentiment and SVM for predicting customer response sentiment. Expert Syst Appl. 2020;162:113746, doi:10.1016/j.eswa.2020.113746
[20] Newman H, Joyner D. Sentiment analysis of student evaluations of teaching. Paper Presented at the International Conference on Artificial Intelligence in Education; 2018.
[21] Ning X, Yac L, Wang X, et al. Rating prediction via generative convolutional neural networks based regression. Pattern Recognit Lett. 2020;132:12–20. doi:10.1016/j.patrec.2018.07.028
[22] Seo S, Huang J, Yang H, et al. Representation learning of users and items for review rating prediction using attention-based convolutional neural network. Paper Presented at the International Workshop on Machine Learning Methods for Recommender Systems; 2017.
[23] Sharma RD, Tripathi S, Sahu SK, et al. Predicting online doctor ratings from user reviews using convolutional neural networks. Int J Mach Learn Comput. 2016;6(2):149, doi:10.18178/ijmlc.2016.6.2.590
[24] Zhao G, Qian X, Xie X. User-service rating prediction by exploring social users' rating behaviors. IEEE Trans Multimed. 2016;18(3):496–506. doi:10.1109/TMM.2016.2515362
[25] Huang F, Li X, Yuan C, et al. Attention-emotion-enhanced convolutional LSTM for sentiment analysis. IEEE Trans Neural Netw Learn Syst. 2021.
[26] Forouzandeh S, Berahmand K, Rostami M. Presentation of a recommender system with ensemble learning and graph embedding: a case on MovieLens. Multimed Tools Appl. 2021;80(5):7805–7832. doi:10.1007/s11042-020-09949-5
[27] Long L, Yin Y, Huang F. Graph-Aware collaborative filtering for Top-N recommendation. Paper Presented at the 2021 International Joint Conference on Neural Networks (IJCNN); 2021.
[28] Shams B, Haratizadeh S. Graph-based collaborative ranking. Expert Syst Appl. 2017;67:59–70. doi:10.1016/j.eswa.2016.09.013
[29] Zhang M, Chen Y. (2019). Inductive matrix completion based on graph neural networks. *arXiv preprint arXiv:1904.12058.*
[30] Chambua J, Niu Z, Yousif A, et al. Tensor factorization method based on review text semantic similarity for rating prediction. Expert Syst Appl. 2018;114:629–638. doi:10.1016/j.eswa.2018.07.059
[31] Margaris D, Vassilakis C. Improving collaborative filtering's rating prediction accuracy by considering users' rating variability. Paper presented at the 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech); 2018.
[32] Huang F, Zhang S, Zhang J, et al. Multimodal learning for topic sentiment analysis in microblogging. Neurocomputing. 2017;253:144–153. doi:10.1016/j.neucom.2016.10.086
[33] Khan ZY, Niu Z, Sandiwarno S, et al. Deep learning techniques for rating prediction: a survey of the state-of-the-art. Artif Intell Rev. 2021;54(1):95–135. doi:10.1007/s10462-020-09892-9
[34] Tan Y, Zhang M, Liu Y, et al. Rating-boosted latent topics: Understanding users and items with ratings and reviews. Paper Presented at the IJCAI; 2016.
[35] Barbosa RRL, Sánchez-Alonso S, Sicilia-Urban MA. Evaluating hotels rating prediction based on sentiment analysis services. Aslib J Inf Manage. 2015.
[36] Chen R-C. User rating classification via deep belief network learning and sentiment analysis. IEEE Trans Comput Social Syst. 2019;6(3):535–546. doi:10.1109/TCSS.2019.2915543
[37] Gojali S, Khodra ML. Aspect based sentiment analysis for review rating prediction. Paper presented at the 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA); 2016.
[38] Lei X, Qian X, Zhao G. Rating prediction based on social sentiment from textual reviews. IEEE Trans Multimedia. 2016;18(9):1910–1921. doi:10.1109/TMM.2016.2575738
[39] Ma X, Lei X, Zhao G, et al. Rating prediction by exploring user's preference and sentiment. Multimed Tools Appl. 2018;77(6):6425–6444. doi:10.1007/s11042-017-4550-z
[40] Cheng Z, Ding Y, Zhu L, et al. Aspect-aware latent factor model: Rating prediction with ratings and reviews. Paper Presented at the Proceedings of the 2018 World Wide web Conference; 2018.
[41] Bhatt A, Patel A, Chheda H, et al. Amazon review classification and sentiment analysis. Int J Comput Sci Inf Technol. 2015;6(6):5107–5110.
[42] Haque TU, Saber NN, Shah FM. Sentiment analysis on large scale Amazon product reviews. Paper Presented at the 2018 IEEE International Conference on Innovative Research and Development (ICIRD); 2018.
[43] Kumar S, De K, Roy PP. Movie recommendation system using sentiment analysis from microblogging data. IEEE Trans Comput Social Syst. 2020;7(4):915–923. doi:10.1109/TCSS.2020.2993585
[44] Kumar S, Gahalawat M, Roy PP, et al. Exploring impact of age and gender on sentiment analysis using machine learning. Electronics (Basel). 2020;9(2):374, doi:10.3390/electronics9020374
[45] Mukherjee A, Mukhopadhyay S, Panigrahi PK, et al. Utilization of oversampling for multiclass sentiment analysis on Amazon review dataset. Paper Presented at the 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST); 2019.
[46] Pathak A, Kumar S, Roy PP, et al. Aspect-Based sentiment analysis in hindi language by ensembling Pre-trained mBERT models. Electronics (Basel). 2021;10(21):2641, doi:10.3390/electronics10212641
[47] Rain C. Sentiment analysis in Amazon reviews using probabilistic machine learning. Swarthmore College; 2013.

[48] Ni J, Li J, McAuley J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. Paper Presented at the Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019.

[49] Cai C, He R, McAuley J. (2017). SPMC: socially-aware personalized markov chains for sparse sequential recommendation. *arXiv preprint arXiv:1708.04497*.