DEVELOPMENT OF ANALYSIS OF THE ACCURACY AND THE DIVERSITY IN THE RECOMMENDER SYSTEM BASED ON COLLABORATIVE FILTERING APPROACH

Vivat Thongchotchat

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Information Technology
Graduate School
Thai-Nichi Institute of Technology
Academic Year 2019

Thesis Topic	Development of Analysis of the Accuracy and the Diversity in the Recommender System based on Collaborative Filtering Approach
By	Vivat Thongchotchat
Field of Study	Information Technology
Thesis Advisor	Dr. Nattagit Jiteurtragool

The Graduates School of Thai-Nichi Institute of Technology has been approved and accepted as partial fulfillment of the requirement for the Master's Degree

..... Dean of the Graduate School
(Assoc. Prof. Dr. Pichit Sukcharoenpong)
Month…..…... Date…..... Year…..

Thesis Committee

..... Chairperson
(Dr. Thongchai KeawKiriya)

..... Committee
(Asst. Dr. Prajak Chertchom)

..... Committee
(Dr. Chadaporn Keatmanee)

..... Advisor
(Dr. Nattagit Jiteurtragool)

VIVAT THONGCHOTCHAT : DEVELOPMENT OF ANALYSIS OF THE ACCURACY AND THE DIVERSITY IN THE RECOMMENDER SYSTEM BASED ON COLLABORATIVE FILTERING APPROACH. ADVISOR : DR. NATTAGIT JITEURTRAGOOL, 112 PP.

One of the current challenges for improving recommender systems is to find an optimal way for diversity and accuracy trade-off. This research aims to find that for real-life educational data, how much impact of diversity for accuracy of the system by developing the collaborative filtering recommender system to conduct experiments and analysis. An analysis showed that using MSD similarity as the similarity calculation method and KNN with Means as the algorithm will give the best prediction result for the user-based system. For the item-based system, using cosine similarity as the similarity calculation method and KNN Baseline as the algorithm will give the best prediction result. Diversity of recommended item affects each system differently. For user-based system, having more choices for the recommended subjects can lead to better prediction result due to there is more data that can be used. For the item-based system, having more choices for a recommended item may not lead to better prediction result due to similar recommended item cannot be used much in the item-based system.

Graduate School Student's Signature……...

Field of Study Information Technology Advisor's Signature……...

Academic Year 2019

Acknowledgement

The author wishes to express gratitude and respect for the dedication of this thesis from family, and friends. The author gives the most grateful to Dr. Nattagit Jiteurtrakool, Ass. Prof. Prajak Chertchom, and Prof. Dr. Sato Kazuhiko, Muroran Institute of Technology, for many supports and opportunities throughout this study.

Subsequently, grateful acknowledgment to Dr. Thongchai KeawKiriya and Dr. Chadaporn Keatmanee, thesis committees, for many suggestions. The author wishes to let this thesis set as a good reference for interested to be used further in the research and project.

Vivat Thongchotchat

List of Content

List of Content (Continued)

List of Tables

List of Tables (Continued)

List of Tables (Continued)

List of Tables (Continued)

List of Tables (Continued)

List of Tables (Continued)

List of Tables (Continued)

List of Figures

Chapter 1
Introduction

1.1 Background

Every day, people are involved with choices and options [1]. What clothes to wear? What movie to watch? What book to read? The sizes of these decision domains are frequently massive, for example, Netflix, one of the most famous TV programs provider, has over 17,000 movies in its programs selection [2], and Amazon.com, one of the most famous online retailer, has over 410,000 titles in its online book store, Kindle, alone [3]. People always rely on recommendations from their friends or experts to support their decisions and to discover new things but these methods of recommending have their limits, particularly for information discovery. There may be an independent film or book that a person would enjoy, but no one of their friends has heard of it yet. To support discovery in this vast size of nowadays information is quite challenging. Even simple decisions like what movie should I see this long weekend can be a difficult decision. As the internet began to develop, data and information emerged rapidly every second. The explosive growth and variety of information available frequently overwhelmed people, leading them to make poor decisions. Many people were finding it difficult to arrive at the most appropriate decision from the vast variety of options. The availability of choices, instead of producing a benefit, started to disrupt people's ability to make a decision. It was understood that while choice is good, more choice is not always better. Indeed, choice, with its suggestions can become excessive, creating a sense that free will to choose may come to be regarded as a kind of misery-inducing tyranny [4].

The recommender system has been recently proved to be an effective method for dealing with too much data and information problems. The recommender system addresses this problem by pointing a user towards new, not-yet-experienced items that may be relevant to the users' recent task. The system generates a recommendation using various types of knowledge and data about the user, the available items, and previously stored transactions. The study of recommender systems is relatively new compared to

research into other classical information system tools and techniques. Recommender systems emerged as an independent research area in the mid-1990s [5, 6, 7, 8].

There are several domains that the recommender system can be used in, education is one of those. There are many pieces of research on how to use the recommender system to solve challenges and issues in the field of education. C. Vialardi-Sacín et al. [9] found that one of the main problems faced by university undergraduates was to make the right decision in relation to their academics itinerary based on the available information (for example courses, schedules, sections, classrooms, and professors). The research proposed the use of the recommender system based on data mining techniques to help undergraduates made decisions on their academic itineraries. More specifically, it provided support to better choose how many and which courses to enroll in, having as basis the experience of the previous undergraduates with similar academic achievements. By analyzing real data corresponding to seven years of student enrollment at the school of system engineering at Universidad de Lima. Based on the analysis, the recommender system was developed. In the research, the data of enrollments was composed of demographic information of each undergraduate, enrollment in course, grade obtained, number of courses taken at each academic term, average grade and cumulative grade per academic term. After filtering and cleaning the data, the learning algorithm C4.5 [10] was applied to obtain rules that are used for the system to suggest the undergraduate if his/her enrollment in a certain course has good probabilities of success or not [11].

K.I. Ghauth and N.A. Abdullah [12] stated that the enormous number of learning materials in e-learning had led to the difficulty of finding suitable learning materials for a particular learning topic, creating the need for recommendation tools within a learning context. The research aimed to propose a novel e-learning recommender system framework that was based on two conceptual foundations, peer learning and social learning theories that encourage students to cooperate and learn among themselves. The proposed framework works on the idea of recommending learning materials with similar content and indicating the quality of learning materials based on good learners' ratings. Proposed e-learning recommender system developed by combining two types of recommendation systems, namely: (i) content-based recommendation and (ii) recommendation based on good learners' ratings. The

objective of the first recommendation type was to recommend the additional learning resources that are similar to those of the viewing item. It ensured that the recommended items remained within the learning context. The second recommendation type aimed to guide learners in selecting good learning resources in order to improve their understanding of the learning topic. The terms ‘‘good learners’’ and ‘‘items’’ was used in the research was defined as follows. Good learners were the learners who have studied the learning materials, and completed the post-test evaluation and achieved marks above 80%. Items or learning materials can be divided into chapters or sub-chapters and accompanied by the item attributes. Item attributes consist of author, title, and keywords.

Commonly, the recommender system is designed to return a number of similar cases in order to provide the user with choices of recommendation. For example, travel recommender such as TripAdvisor typically returns the k best cases such as holiday packages or apartment listings for user recommended lists. The objective is to satisfy user needs with a single search with the retrieval of multiple cases and to maximize the likelihood of relevant cases appearing high up in the result list with the priority given to similarity. However the similarity-based method is not the best choice in some cases, for example, in case of travel recommender: the user submits the request for a 2-days weekend vacation in Okinawa, costing less than ￥200,000, within 4 hours flying time, and with good night-life and famous restaurant nearby. The best choice that the system recommended is the hotel in the Kokusai Doori area for the first weekend in March. A good recommendation, but what if the second, third, and fourth recommendations are from the same area? While the k best recommendations are all very similar to the target request, they are also very similar to each other. The user will not have received a useful set of alternatives if the first recommendation is unsuitable. For instance, in this example, if the user decides to avoid the Kokusai Doori area, then none of the alternative recommendations will be satisfied and he/she will have to initiate a new search. Even though precisely predicting the users’ interests was the main objective of the recommender systems field for a long time since the beginning of the field’s development, other perspectives toward recommendation utility besides prediction accuracy, started to appear in the literature by the beginning of the 2000s [13, 14], taking views that began to realize the importance of novelty and diversity, among other

properties, in the added value of recommendation [15, 16]. This realization grew progressively, reaching an upswing of activity by the turn of the past decade [17, 18, 19, 20, 21]. Nowadays, it might be said that novelty and diversity are becoming an increasingly frequent part of evaluation practice. They are being included increasingly often among the reported effectiveness metrics of new recommendation approaches, and are explicitly targeted by algorithmic innovations time and again. And it seems difficult to conceive progress in the recommender systems field without considering these dimensions.

In this research, the recommender system was developed in order to discover the relationship between the accuracy of the recommender system and the diversity of recommended items based on the real data in the field of education. Datasets from the Muroran Institute of Technology was used to develop the recommender system. This data was the real data which means it consists of many biases, outliers, and missing values so the analysis would reflect the real relation that one who wants to develop the recommender system for use in the field of education needs to be concerned. This data was the academic history data of each undergraduate from the faculty of information technology between Heisei 18 academic year (2006 A.C.) and Heisei 20 academic year (2008 A.C.). For each academic year, each undergraduate needs to enroll for a subject in order to collect enough credits that need to be fulfilled to reach the threshold for graduation. Among the subject that undergraduates can enroll are divided into two main groups; compulsory subjects and elective subjects. The compulsory subjects are subjects that each undergraduate needs to enroll and pass in order to graduate. Every undergraduate need to take this subject so it cannot be used as the recommended items for the system because it is no use to recommend this group of a subject as everyone needs to enroll it anyway. The elective courses are the one that will be focused on as the recommended item for the system as it keeps changing every academic year make it fit perfectly to be used in the analysis because, in each academic year, a number of subjects will be changed due to many reasons such as the retirement of lecturer so the diversity of recommended item or in this context, the subject, will become more varied as time passes and will be used to develop the recommender system that will give a recommendation by predicting what score will receive if he/she enrolls in that recommend subject. After that, the analysis for comparison between the diversity of

subjects that will be different in each academic year and the accuracy of the score that the system predicts was conducted.

1.2 Objectives

1.2.1 To analyze the relation between the diversity and the accuracy of recommendation system based on real-world data in the field of education.

1.2.2 To discover the base knowledge that can be set as a standard or a case study for developing the recommendation system in the field of education

1.3 Boundaries

1.3.1 Boundary of data

This study will only use datasets from Muroran Institute of Technology and only the academic history data from undergraduates that enroll between academic year Heisei 18 (2006 A.C.) and Heisei 20 (2008 A.C.).

1.3.2 Boundary of system

This study will only use collaborative filtering method for developing the system.

1.3.3 Boundary of evaluation

This study will only use root mean square error as the evaluation method for accuracy of the prediction.

1.4 Benefits

1.4.1 Knowledge of the relation between accuracy and diversity that can be used to further develop the recommendation system in the education field.

1.5 Definition

1.5.1 **Recommender System**

Software tools and techniques that provide suggestions for items that are most likely of interest to a particular user. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read. “Item” is the general term used to denote what the system recommends to users. A recommendation system normally focuses on a specific type of item (e.g., CDs or

news) and according to its design, its graphical user interface, and the core recommendation technique used to generate the recommendations are all customized to provide useful and effective suggestions for that specific type of item. For example, system that suggest the item that other customer that bought this item also bought from Amazon.

1.5.2 **Python**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. [22]

1.5.3 **Package**

In order to understand “package” in Python, the word “module” is needed to understand first. If you quit from the Python interpreter and enter it again, the definitions you have made (functions and variables) are lost. Therefore, if you want to write a somewhat longer program, you are better off using a text editor to prepare the input for the interpreter and running it with that file as input instead. This is known as creating a script. As your program gets longer, you may want to split it into several files for easier maintenance. You may also want to use a handy function that you’ve written in several programs without copying its definition into each program.

To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a module; definitions from a module can be imported into other modules or into the main module (the collection of variables that you have access to in a script executed at the top level and in calculator mode).

Packages are a way of structuring Python’s module namespace by using “dotted module names”. For example, the module name A.B designates a submodule named B in a package named A. Just like the use of modules saves the authors of

different modules from having to worry about each other's global variable names, the use of dotted module names saves the authors of multi-module packages like NumPy or Pillow from having to worry about each other's module names. [23]

1.5.4 **Google Colaboratory**

A research tool for machine learning education and research. It's a Jupyter notebook environment that requires no setup to use. [24]

1.5.5 **Surprise Package**

Surprise is a Python scikit building and analyzing recommender systems that deal with explicit rating data. Surprise was designed with the following purposes in mind:

1.5.5.1 Give users perfect control over their experiments. To this end, a strong emphasis is laid on documentation, which we have tried to make as clear and precise as possible by pointing out every detail of the algorithms.

1.5.5.2 Alleviate the pain of Dataset handling. Users can use both built-in datasets (Movielens, Jester), and their own custom datasets.

1.5.5.3 Provide various ready-to-use prediction algorithms such as baseline algorithms, neighborhood methods, matrix factorization-based (SVD, PMF, SVD++, NMF), and many others. Also, various similarity measures (cosine, MSD, pearson…) are built-in.

1.5.5.4 Make it easy to implement new algorithm ideas.

1.5.5.5 Provide tools to evaluate, analyse and compare the algorithms performance. Cross-validation procedures can be run very easily using powerful CV iterators (inspired by scikit-learn excellent tools), as well as exhaustive search over a set of parameters.

Chapter 2
Literature Review

This research involved the following idea and theory. First, the definition of the recommender system along with some specific words that are used in this field will be explained. Second, Collaborative Filtering, one of the methods used in developing the recommender system will be explained. Third, the novelty and diversity, the research topic that will be analyzed in this study, will be explained. Lastly, some of the relevant research about similarity and diversity in the recommender system will be explained.

2.1 Recommender System

Recommender Systems are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user [7, 25, 26]. The suggestions relating to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read. “Item” is the common term used to represent what the system recommends to users. The recommender system normally focuses on a specific type of item according to its design, its graphical user interface, and its core recommendation technique used to generate the recommendations.

The recommender system is basically designed for a person who lacks sufficient personal experience or relevant knowledge in order to provide potential choices from the overwhelming number of alternative items [7]. A good example is the item recommender system that assists users in shopping which was deployed on one of the well-known retailer websites, Amazon.com for the purpose to personalize the online store for each customer [27]. Since recommendations are usually personalized, different users or user groups benefit from diverse, tailored suggestions. The system tries to predict what the most suitable products or services are, based on the user’s preferences and constraints. In order to complete such a computational task, the system collects information from users regarding their preferences which are either explicitly expressed, e.g., as satisfaction for products or are inferred by interpreting the actions of the user. For instance, a recommender system may consider the navigation to a

particular product page as an implicit sign of preference for the items shown on that page.

The development of the recommender system initiated from a rather simple observation: individuals often rely on recommendations provided by others in making routine, daily-decisions [7, 8]. For example, it is common to rely on what one's peers recommend when selecting a book to read; employers count on recommendation letters in their recruiting decisions; and when selecting a movie to watch, individuals tend to read and rely on the movie reviews that a film critic has written, which appear in the newspaper they read. In seeking to mimic this behavior, the first recommender system applied algorithms in order to leverage recommendations produced by a community of users and deliver these recommendations to an "active" user, or a user looking for suggestions. The recommendations were for items that similar users, or those with similar tastes, had liked. This approach is termed collaborative-filtering and its rationale follows that if the active user agreed in the past with certain users, then the other recommendations coming from these similar users should be relevant as well as of interest to the active user.

Recommender systems are information processing systems that actively gather various kinds of data in order to build their recommendations. Data is primarily about the items to suggest and the users who will receive these recommendations. But, since the data and knowledge sources available for recommender systems can be very diverse, ultimately, whether it can be exploited or not depends on the recommendation technique. In general, there are recommendation techniques that are knowledge-poor, namely, that use very simple and basic data, such as user ratings or evaluations for items. Other techniques are much more knowledge-dependent, in that they use ontological descriptions of the users or the items, constraints, or social relations and activities of the user. In any case, as a general classification, data used by the recommender system refers to three kinds of objects: items, users, and transactions, that is, the relations between the users and the items.

Items : Items are the objects that are recommended. Items may be characterized by their complexity and their value or utility. The value of an item may be positive if the item is useful to the user, or negative if the item is not appropriate and the user made the wrong decision when selecting it. We note that when a user is acquiring an item,

one will always incur a cost which includes the cognitive cost of searching for the item and the real monetary cost eventually paid for the item.

For instance, the designer of a news recommender system must take into account the complexity of a news item, i.e., its structure, the textual representation, and the time-dependent importance of any news item. But at the same time, recommender system designers must understand that even if the user is not paying for reading news, there is always a cognitive cost associated with searching and reading news items. If a selected item is relevant to the user, this cost is dominated by the benefit of having acquired useful information. Whereas if the item is not relevant, the net value of that item for the user, and its recommendation, is negative. In other domains, e.g., cars, or financial investments, the true monetary cost of the items becomes an important element to consider when selecting the most appropriate recommendation approach.

Items with low complexity and value are news, webpages, books, CDs, and movies. Items with larger complexity and value are digital cameras, mobile phones, PCs, etc. The most complex items that have been considered are insurance policies, financial investments, travel, and jobs [28]. Recommender systems, according to their core technology, can use a range of properties and features of the items. For example in a movie recommender system, the genre (comedy, thriller, etc.), as well as the director and actors, can be used to describe a movie and to learn how the utility of an item depends on its features. Items can be represented using various information and representation approaches, e.g., in a minimalist way as a single ID code, or in a richer form, as a set of attributes, and even as a concept in an ontological representation of the domain.

Users : Users of an RS, as mentioned above, may have very diverse goals and characteristics. In order to personalize the recommendations and the human-computer interaction, recommender systems exploit a range of information about the users. This information can be structured in various ways, and again, the selection of what information to model depends on the recommendation technique.

For instance, in collaborative filtering, users are modeled as a simple list containing the ratings provided by the user for certain items. In a demographic RS, sociodemographic attributes such as age, gender, profession, and education, are used. User data is said to constitute the user model [29, 30]. The user model profiles the user,

i.e., encodes her preferences and needs. Various user modeling approaches have been used and, in a certain sense, an RS can be viewed as a tool that generates recommendations by building and exploiting user models [31, 32]. Since no personalization is possible without a convenient user model the user model will always play a central role. For instance, in reconsidering a collaborative filtering approach, the user is either profiled directly by its ratings of items or, using these ratings, the system derives a vector of factor values where users differ in how each factor weighs in their model.

Users can also be described by their behavior pattern data, for example, site browsing patterns (in a Web-based recommender system) [33], or travel search patterns (in a travel recommender system) [34]. Moreover, user data may include relations between users such as the trust level of these relations between users. An RS might utilize this information to recommend items to users that were preferred by similar or trusted users.

Transactions : It was generically referred to as a transaction as a recorded interaction between a user and the RS. Transactions are log-like data that store important information generated during the human-computer interaction and which are useful for the recommendation generation algorithm that the system is using. For instance, a transaction log may contain a reference to the item selected by the user and a description of the context (e.g., the user goal/query) for that particular recommendation. If available, that transaction may also include explicit feedback that the user has provided, such as the rating for the selected item.

In fact, the ratings are the most popular form of transaction data that an RS collects. These ratings may be collected explicitly or implicitly. In the explicit collection of ratings, the user is asked to provide an opinion about an item on a rating scale. According to [35], ratings can take on a variety of forms:

- Numerical ratings such as the 1–5 stars provided in the book recommender associated with Amazon.com.
- Ordinal ratings, such as “strongly agree, agree, neutral, disagree, strongly disagree” where the user is asked to select the term that best indicates his or her opinion regarding an item (usually via questionnaire).

● Binary ratings that model choices in which the user is simply asked to decide if a certain item is good or bad.

● Unary ratings can indicate that a user has observed or purchased an item, or otherwise rated the item positively. In such cases, the absence of a rating indicates that we have no information relating to the user to the item (perhaps the user purchased the item elsewhere).

Another form of user evaluation consists of tags associated with the user with the items that the system presents. For instance, on Movielens), recommender system tags represent how MovieLens users feel about a movie, e.g.: “too long,” or “acting.” In transactions that collect implicit ratings, the system aims to infer the user’s opinion based on the user’s actions. For example, if a user enters the keyword “Yoga” at Amazon.com, a long list of books will be provided. In return, the user may click on a certain book on the list in order to receive additional information. At this point, the system may infer that the user is somewhat interested in that book.

In order to implement its core function, identifying useful items for the user, a recommender system must predict that an item is worth recommending. In order to do this, the system must be able to predict the utility of some items, or at least compare the utility of some items, and then decide which items to recommend based on this comparison. The prediction step may not be explicit in the recommendation algorithm but we can still apply this unifying model to describe the general role of a recommender system. Here, our goal is to provide the reader with a unifying perspective rather than an account of all the different recommendation approaches that will be illustrated in this handbook.

To illustrate the prediction step of recommender systems, consider, for instance, a simple and non-personalized recommendation algorithm that recommends only the most popular songs. The rationale for using this approach is that in the absence of more precise information about the user’s preferences, a popular song, i.e., one that is liked (high utility) by many users, will also most-likely appeal to a generic user, or at least with a higher likelihood than another randomly selected song. Hence, the utility of such popular songs is predicted to be reasonably high for this generic user.

This view of the core recommendation computation as the prediction of the utility of an item for a user has been suggested in [36] and recently updated in [37].

Both papers model this degree of utility of the user u for the item i as a (real-valued) function $R(u,i)$ as is normally done in collaborative filtering by considering the ratings of users for items. Then, the fundamental task of a collaborative filtering recommender system is to predict the value of R over pairs of users and items, or in other words, to compute $\hat{R}(u,i)$, where we denote with $\hat{R}$ the estimation, computed by the recommender system, of the true function R. Consequently, having computed this prediction for the active user u on a set of items, i.e., $\hat{R}(u,i_1)$,...,$\hat{R}(u,i_N)$, the system will recommend the items i_{j1},...,i_{jk} (K ≤ N) with the largest predicted utility. K is typically a small number, that is, much smaller than the cardinality of the item data set or the items on which a user utility prediction can be computed, i.e., recommender system "filter" the items that are recommended to users.

As mentioned above, some recommender systems do not fully estimate the utility before making a recommendation, but they may apply some heuristics to hypothesize that an item may be of use to a user. This is typical, for instance, in knowledge-based systems. These utility predictions are computed with specific algorithms and use various kinds of knowledge about users, items, and the utility function itself [25]. For instance, the system may assume that the utility function is Boolean and therefore it will just determine whether an item is or is not useful for the user. Consequently, assuming that there is some available knowledge, or possibly none, about the user who is requesting the recommendation, as well as knowledge about items, and other users who received recommendations, the system will leverage this knowledge with an appropriate algorithm to generate various utility predictions and hence recommendations [25].

It is also important to note that sometimes the user utility for an item is observed to depend on other variables, which we generically call "contextual" [37]. For instance, the utility of an item for a user can be influenced by the domain knowledge of the user (e.g., expert versus beginning users of a digital camera), or can depend on the time when the recommendation is requested. Equally, users may be more interested in items (e.g., restaurant) closer to their current location. Consequently, the recommendations must be adapted to these specific additional details and as a result, it becomes increasingly more difficult to correctly estimate what the right recommendations are.

Item recommendation approaches can be divided into two broad categories: personalized and non-personalized. Among the personalized approaches are content-based and collaborative filtering methods, as well as hybrid techniques combining these two types of methods. The general principle of content-based (or cognitive) methods [38, 39, 40, 41] is to identify the common characteristics of items that have received a favorable rating from a user and then recommend to this user's new item that shares these characteristics. Recommender systems based purely on content generally suffer from the problems of limited content analysis and over-specialization [8]. Limited content analysis occurs when the system has a limited amount of information on its users or the content of its items. For instance, privacy issues might refrain a user from providing personal information, or the precise content of items may be difficult or costly to obtain for some types of items, such as music or images. Another problem is that the content of an item is often insufficient to determine its quality. Over-specialization, on the other hand, is a side effect of the way in which content-based systems recommend new items, where the predicted rating of a user for an item is high if this item is similar to the ones liked by this user. For example, in a movie recommendation application, the system may recommend to a user a movie of the same genre or having the same actors as movies already seen by this user. Because of this, the system may fail to recommend items that are different but still interesting to the user.

2.2 Collaborative Filtering

Collaborative filtering (CF) [42] is a popular recommendation algorithm that bases its predictions and recommendations on the ratings or behavior of other users in the system. The fundamental assumption behind this method is that other users' opinions can be selected and aggregated in such a way as to provide a reasonable prediction of the active user's preference. Intuitively, they assume that, if users agree about the quality or relevance of some items, then they will likely agree about other items if a group of users likes the same things as Mary, then Mary is likely to like the things they like which she hasn't yet seen. Collaborative (or social) filtering approaches use the rating information of other users and items in the system. The key idea is that the rating of a target user for a new item is likely to be similar to that of another user if both users have rated other items in a similar way. Likewise, the target user is likely to

rate two items in a similar fashion, if other users have given similar ratings to these two items. Collaborative approaches overcome some of the limitations of content-based ones. For instance, items for which the content is not available or difficult to obtain can still be recommended to users through the feedback of other users. Furthermore, collaborative recommendations are based on the quality of items as evaluated by peers, instead of relying on content that may be a bad indicator of quality. Finally, unlike content-based systems, collaborative filtering ones can recommend items with very different content, as long as other users have already shown interest in these different items.

Collaborative filtering approaches can be grouped into two general classes of the neighborhood and model-based methods. In neighborhood-based (memory-based [43] or heuristic-based [36]) collaborative filtering [7, 8, 27, 44, 45, 46, 47, 48, 49], the user-item ratings stored in the system are directly used to predict ratings for new items. This can be done in two ways known as user-based or item-based recommendations. User-based systems, such as GroupLens [47], Bellcore video [46], and Ringo [8], evaluate the interest of a target user for an item using the ratings for this item by other users, called neighbors, that have similar rating patterns. The neighbors of the target users are typically the users whose ratings are most correlated to the target user's ratings. Item-based approaches [27, 45, 49], on the other hand, predict the rating of a user for an item based on the ratings of the user for similar items. In such approaches, two items are similar if several users of the system have rated these items in a similar fashion.

While recent investigations show state-of-the-art model-based approaches superior to neighborhood ones in the task of predicting ratings [50, 51], there is also an emerging understanding that good prediction accuracy alone does not guarantee users an effective and satisfying experience [10]. Another factor that has been identified as playing an important role in the appreciation of users for the recommender system is serendipity [49]. Serendipity extends the concept of novelty by helping a user find an interesting item he or she might not have otherwise discovered. For example, recommending to a user a movie directed by his favorite director constitutes a novel recommendation if the user was not aware of that movie, but is likely not serendipitous since the user would have discovered that movie on his own.

Model-based approaches excel at characterizing the preferences of a user with latent factors. For example, in a movie recommender system, such methods may determine that a given user is a fan of movies that are both funny and romantic, without having to actually define the notions “funny” and “romantic”. This system would be able to recommend to the user a romantic comedy that may not have been known to this user. However, it may be difficult for this system to recommend a movie that does not quite fit this high-level genre, for instance, a funny parody of horror movies. Neighborhood approaches, on the other hand, capture local associations in the data. Consequently, it is possible for a movie recommender system based on this type of approach to recommend the user a movie very different from his usual taste or a movie that is not well known (e.g. repertoire film), if one of his closest neighbors has given it a strong rating. This recommendation may not be a guaranteed success, as would be a romantic comedy, but it may help the user discover a whole new genre or a new favorite actor/director.

The reason collaborative filtering or neighborhood-based was used as the method for making a recommendation are:

● Simplicity: Neighborhood-based methods are intuitive and relatively simple to implement. In their simplest form, only one parameter (the number of neighbors used in the prediction) requires tuning.

● Justifiability: Such methods also provide a concise and intuitive justification for the computed predictions. For example, in item-based recommendation, the list of neighbor items, as well as the ratings given by the user to these items, can be presented to the user as a justification for the recommendation. This can help the user better understand the recommendation and its relevance, and could serve as the basis for an interactive system where users can select the neighbors for which greater importance should be given in the recommendation.

● Efficiency: One of the strong points of neighborhood-based systems is their efficiency. Unlike most model-based systems, they require no costly training phases, which need to be carried at frequent intervals in large commercial applications. These systems may require pre-computing nearest neighbors in an offline step, which is typically much cheaper than model training, providing near-instantaneous recommendations. Moreover, storing these nearest neighbors requires very little

memory, making such approaches scalable to applications having millions of users and items.

- Stability: Another useful property of recommender systems based on this approach is that they are little affected by the constant addition of users, items, and ratings, which are typically observed in large commercial applications. For instance, once item similarities have been computed, an item-based system can readily make recommendations to new users, without having to re-train the system. Moreover, once a few ratings have been entered for a new item, only the similarities between this item and the ones already in the system need to be computed.

In order to give a formal definition of the item recommendation task, we introduce the following notation. The set of users in the recommender system will be denoted by U, and the set of items by J. Moreover, we denote by R the set of ratings recorded in the system and write S the set of possible values for a rating (e.g., S D = [1; 5] or S = [Like; dislike]. Also, we suppose that no more than one rating can be made by any user $u \in U$ for a particular item $i \in I$ and write r_{ui} this rating. To identify the subset of users that have rated an item i, we use the notation U_i. Likewise, I_u represents the subset of items that have been rated by a user u. Finally, the items that have been rated by two users u and v, i.e. $I_u \cap I_v$, is an important concept in our presentation, and we use I_{uv} to denote this concept. In a similar fashion, U_{ij} is used to denote the set of users that have rated both items i and j.

Two of the most important problems associated with recommender systems are the rating prediction and top-N recommendation problems. The first problem is to predict the rating that a user u will give his or her unrated item i. When ratings are available, this task is most often defined as a regression or (multi-class) classification problem where the goal is to learn a function $f : U \, x \, J \longrightarrow S$ that predicts the rating $f(u, i)$ of a user u for a new item i. Accuracy is commonly used to evaluate the performance of the recommendation method. Typically, the rating R are divided into a training set R_{train} used to learn f, and a test set R_{test} used to evaluate the prediction accuracy. Two popular measures of accuracy are the Mean Absolute Error (MAE):

$$MAE(f) = \frac{1}{|\mathcal{R}_{test}|} \sum_{r_{ui} \in \mathcal{R}_{test}} |f(u,i) - r_{ui}| \quad (2.1)$$

and the Root Mean Squared Error (RMSE):

$$RMSE(f) = \sqrt{\frac{1}{|\mathcal{R}_{test}|} \sum_{r_{ui} \in \mathcal{R}_{test}} (f(u,i) - r_{ui})^2} \quad (2.2)$$

Recommender systems based on neighborhood automate the common principle that similar users prefer similar items, and similar items are preferred by similar users. To illustrate this, consider the following example based on the ratings of table. 2.1.

Example 2.1 User Eric has to decide whether or not to rent the movie "Titanic" that he has not yet seen. He knows that Lucy has very similar tastes when it comes to movies, as both of them hated "The Matrix" and loved "Forrest Gump", so he asks her opinion on this movie. On the other hand, Eric finds out he and Diane have different tastes, Diane likes action movies while he does not, and he discards her opinion or considers the opposite in his decision.

Table 2.1 example showing the ratings of four users for five movies

	The Matrix	Titanic	Die Hard	Forrest Gump	Wall-E
John	5	1		2	2
Lucy	1	5	2	5	5
Eric	2	?	3	5	4
Diane	4	3	5	3	

2.2.1 User-based Collaborative Filtering

User-based neighborhood recommendation methods predict the rating r_{ui} of a user u for a new item i using the ratings given to i by users most similar to u, called nearest-neighbors. Suppose we have for each user $v \neq u$ a value wuv representing the preference similarity between u and v. The k-nearest-neighbors (k-NN) of u, denoted

by $N(u)$ are the k users v with the highest similarity w_{uv} to u. However, only the users who have rated item i can be used in the prediction of r_{ui}, and we instead consider the k users most similar to u that have rated i. We write this set of neighbors as $N_i(u)$. The rating r_{ui} can be estimated as the average rating given to i by these neighbors:

$$\hat{r}_{ui} = \frac{1}{|\mathcal{N}_i(u)|} \sum_{v \in \mathcal{N}_i(u)} r_{vi} \tag{2.3}$$

A problem with (2.3) is that it does not take into account the fact that the neighbors can have different levels of similarity. Consider once more the example of table 2.1. If the two nearest-neighbors of Eric are Lucy and Diane, it would be foolish to consider equally their ratings of the movie "Titanic", since Lucy's tastes are much closer to Eric's than Diane's. A common solution to this problem is to weigh the contribution of each neighbor by its similarity to u. However, if these weights do not sum to 1, the predicted ratings can be well outside the range of allowed values. Consequently, it is customary to normalize these weights, such that the predicted rating becomes

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} r_{vi}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \tag{2.4}$$

In the denominator of (2.4), $|w_{uv}|$ is used instead of w_{uv} because negative weights can produce ratings outside the allowed range. Also, w_{uv} can be replaced by w_{uv}^{α}, where $\alpha > 0$ is an amplification factor [52]. When $\alpha > 1$, as it is most often employed, an even greater importance is given to the neighbors that are the closest to u.

Example 2.2 Suppose we want to use (2.4) to predict Eric's rating of the movie "Titanic" using the ratings of Lucy and Diane for this movie. Moreover, suppose the similarity weights between these neighbors and Eric are respectively 0.75 and 0.15. The predicted rating would be

$$\hat{r} = \frac{0.75\ x\ 5 + 0.15\ x\ 3}{0.75 + 0.15} \simeq 4.67,$$

which is closer to Lucy's rating than to Diane's.

Equation (2.4) also has an important flaw: it does not consider the fact that users may use different rating values to quantify the same level of appreciation for an item. For example, one user may give the highest rating value to only a few outstanding items, while a less difficult one may give this value to most of the items he likes. This problem is usually addressed by converting the neighbors' ratings r_{vi} to normalized ones $h(r_{vi})$ [7], giving the following prediction:

$$\hat{r}_{ui} = h^{-1}\left(\frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} h(r_{vi})}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}\right) \quad (2.5)$$

Note that the predicted rating must be converted back to the original scale, hence the h-1 in the equation.

2.2.2 Item-based Collaborative Filtering

While user-based methods rely on the opinion of like-minded users to predict a rating, item-based approaches [27, 45, 49] look at the ratings given to similar items. Let us illustrate this approach with our toy example.

Example 2.4 Instead of consulting with his peers, Eric instead determines whether the movie "Titanic" is right for him by considering the movies that he has already seen. He notices that people that have rated this movie have given similar ratings to the movie "Forrest Gump" and "Wall-E". Since Eric liked these two movies he concludes that he will also like the movie "Titanic".

This idea can be formalized as follows. Denote by $N_u(i)$ the items rated by user u most similar to item i. The predicted rating of u for i is obtained as a weighted average of the ratings given by u to the items of $N_u(i)$:

$$\hat{r}_{ui} = \frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij} r_{uj}}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|} \tag{2.6}$$

Example 2.5 Suppose our prediction is again made using two nearest-neighbors, and that the items most similar to “Titanic” are “Forrest Gump” and “Wall-E”, with respective similarity weights 0:85 and 0:75. Since ratings of 5 and 4 were given by Eric to these two movies, the predicted rating is computed as

$$\hat{r} = \frac{0.85\ x\ 5 + 0.75\ x\ 4}{0.85 + 0.75} \simeq 4.53.$$

Again, the differences in the users’ individual rating scales can be considered by normalizing ratings with a function h:

$$\hat{r}_{ui} = h^{-1}\left(\frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij} h(r_{uj})}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|}\right) \tag{2.7}$$

Moreover, we can also define an item-based classification approach. In this case, the items j rated by user u vote for the rating to be given to a new item i, and these votes are weighted by the similarity between i and j. The normalized version of this approach can be expressed as follows:

$$\hat{r}_{ui} = h^{-1}\left(\begin{matrix}\arg max \\ r \in \mathcal{S}'\end{matrix} \sum_{j \in \mathcal{N}_u(i)} \delta(h(r_{ui}) = r) w_{ij}\right) \tag{2.8}$$

2.2.3 Components of Neighborhood Methods

We have seen that deciding between a user-based or item-based recommendation approach, can have a significant impact on the accuracy, efficiency, and overall quality of the recommender system. In addition to these crucial attributes, three very important considerations in the implementation of a neighborhood-based recommender system are (1) the normalization of ratings, (2) the computation of the

similarity weights, and (3) the selection of neighbors. This section reviews some of the most common approaches for these three components, describes the main advantages and disadvantages of using each one of them and gives indications on how to implement them.

2.2.3.1 Rating Normalization

When it comes to assigning a rating to an item, each user has its own personal scale. Even if an explicit definition of each of the possible ratings is supplied (e.g., 1 = "strongly disagree", 2 = "disagree", 3 = "neutral", etc.), some users might be reluctant to give high/low scores to items they liked/disliked. Two of the most popular rating normalization schemes that have been proposed to convert individual ratings to a more universal scale are mean-centering and Z-score.

2.2.3.1.1 Mean-Centering

The idea of mean-centering [7, 52] is to determine whether a rating is positive or negative by comparing it to the mean rating. In user-based recommendation, a raw rating r_{ui} is a transformation to a mean-centered one $h(r_{ui})$ by subtracting to r_{ui} the average $\bar{r}_u$ of the ratings given by user u to the items in $\int_u$:

$$h(r_{ui}) = r_{ui} - \bar{r}_u$$

Using this approach the user-based prediction of a rating r_{ui} is obtained as

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv}(r_{vi} - \bar{r}_v)}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \quad (2.9)$$

In the same way, the item-mean-centered normalization of r_{ui} is given by

$$h(r_{ui}) = r_{ui} - \bar{r}_i$$

where $\bar{r}_i$ corresponds to the mean rating given to item i by user in U_i . This normalization technique is most often used in item-based recommendation, where a rating r_{ui} is predicted as:

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij}(r_{uj} - \bar{r}_j)}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|} \quad (2.10)$$

An interesting property of mean-centering is that one can see right away if the appreciation of a user for an item is positive or negative by looking at the sign of the normalized rating. Moreover, the module of this rating gives the level at which the user likes or dislikes the item.

Table 2.2 the user mean-centered ratings of table 2.1

	The Matrix	Titanic	Die Hard	Forrest Gump	Wall-E
John	2.50	-1.50		-0.50	-0.50
Lucy	-2.60	1.40	-1.60	1.40	1.40
Eric	-1.50		-0.50	1.50	0.50
Diane	0.25	-0.75	1.25	-0.75	

Table 2.3 the item mean-centered ratings of table 2.1

	The Matrix	Titanic	Die Hard	Forrest Gump	Wall-E
John	2.00	-2.00		-1.75	-1.67
Lucy	-2.00	2.00	-1.33	1.25	1.33
Eric	-1.00		-0.33	1.25	0.33
Diane	1.00	0.00	1.67	-0.75	

Example 2.6 As shown in table 2.2, although Diane gave an average rating of 3 to the movies “Titanic” and “Forrest Gump”, the user-mean-centered ratings show that her appreciation of these movies is in fact negative. This is because her ratings are high on average, and so, an average rating corresponds to a low degree of appreciation. Differences are also visible while comparing the two types of

mean-centering. For instance, the item-mean-centered rating of the movie “Titanic” is neutral, instead of negative, due to the fact that much lower ratings were given to that movie. Likewise, Diane’s appreciation for “The Matrix” and John’s distaste for “Forrest Gump” is more pronounced in the item-mean-centered ratings.

2.2.3.1.2 Z-Score Normalization

Consider, two users A and B that both have an average rating of 3. Moreover, suppose that the ratings of A alternate between 1 and 5, while those of B are always 3. A rating of 5 given to an item by B is more exceptional than the same rating given by A, and, thus, reflects a greater appreciation for this item. While mean-centering removes the offsets caused by the different perceptions of an average rating, Z-score normalization [53] also considers the spread in the individual rating scales. Once again, this is usually done differently in user-based than in item-based recommendation. In user-based methods, the normalization of a rating r_{ui} divides the user-mean-centered rating by the standard deviation σ_u of the ratings given by user u:

$$h(r_{ui}) = \frac{r_{ui} - \bar{r}_u}{\sigma_u}$$

A user-based prediction of rating r_{ui} using this normalization approach would therefore be obtained as

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} (r_{vi} - \bar{r}_v) / \sigma_v}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \quad (2.11)$$

2.2.3.1.3 Choosing a Normalization Scheme

In some cases, rating normalization can have undesirable effects. For instance, imagine the case of a user that gave only the highest ratings to the items he has purchased. Mean-centering would consider this user as “easy to please” and any rating below this highest rating (whether it is a positive or negative rating) would be considered as negative. However, it is possible that this user is in fact “hard to please” and carefully selects only items that he will like for sure. Furthermore, normalizing a few ratings can produce unexpected results. For example, if a user has

entered a single rating or a few identical ratings, his rating standard deviation will be 0, leading to undefined prediction values. Nevertheless, if the rating data is not overly sparse, normalizing ratings have been found to consistently improve the predictions [53, 54].Comparing mean-centering with Z-score, as mentioned, the second one has the additional benefit of considering the variance in the ratings of individual users or items. This is particularly useful if the rating scale has a wide range of discrete values or if it is continuous. On the other hand, because the ratings are divided and multiplied by possibly very different standard deviation values, Z-score can be more sensitive than mean-centering and, more often, predict ratings that are outside the rating scale. Lastly, while an initial investigation found mean-centering and Z-score to give comparable results [53], a more recent one showed Z-score to have more significant benefits [54].

Finally, if rating normalization is not possible or does not improve the results, another possible approach to remove the problems caused by the rating scale variance is preference-based filtering. The particularity of this approach is that it focuses on predicting the relative preferences of users instead of absolute rating values. Since an item preferred to another one remains so regardless of the rating scale, predicting relative preferences removes the need to normalize the ratings. More information on this approach can be found in [55, 56, 57, 58].

2.2.3.2 Similarity Weight Computation

The similarity weights play a double role in neighborhood-based recommendation methods: (1) they allow to select trusted neighbors whose ratings are used in the prediction, and (2) they provide the means to give more or less importance to these neighbors in the prediction. The computation of the similarity weights is one of the most critical aspects of building a neighborhood-based recommender system, as it can have a significant impact on both its accuracy and its performance.

2.2.3.2.1 Correlation-Based Similarity

A measure of the similarity between two objects a and b, often used in information retrieval, consists in representing these objects in the form of a vector x_a and x_b and computing the Cosine Vector (CV) (or Vector Space) similarity [38, 39, 40] between these vectors:

$$\cos(x_a, x_b) = \frac{x_a^T x_b}{||x_a|| ||x_b||}$$

In the context of item recommendation, this measure can be employed to compute user similarities by considering a user u as a vector $x_u \in R^{|I|}$, where $x_{ui} = r_{ui}$ if user u has rated item i, and 0 otherwise. The similarity between two users u and v would then be computed as where I_{uv} once more denotes the items rated by both u and v. A problem with this measure is that it does not consider the differences in the mean and variance of the ratings made by users u and v.

$$CV(u, v) = \cos(x_u, x_v) = \frac{\sum_{i \in \mathcal{J}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{J}_u} r_{ui}^2 \sum_{j \in \mathcal{J}_v} r_{vj}^2}} \quad (2.18)$$

A popular measure that compares ratings where the effects of mean and variance have been removed is the Pearson Correlation (PC) similarity:

$$PC(u, v) = \frac{\sum_{i \in \mathcal{J}_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{J}_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in \mathcal{J}_{uv}} (r_{vi} - \bar{r}_v)^2}} \quad (2.19)$$

Note that this is different from computing the CV similarity on the Z-score normalized ratings since the standard deviation of the ratings is evaluated only on the common items I_{uv}, not on the entire set of items rated by u and v, i.e. I_u and I_v. The same idea can be used to obtain similarities between two items i and j [45, 49], this time by comparing the ratings made by users that have rated both these items:

$$PC(i, j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_i)^2 \sum_{u \in \mathcal{U}_{ij}} (r_{uj} - \bar{r}_j)^2}} \quad (2.20)$$

While the sign of a similarity weight indicates whether the correlation is direct or inverse, its magnitude (ranging from 0 to 1) represents the strength of the correlation.

Example 2.7 The similarities between pairs of users and items of our toy example, as computed using PC similarity, are shown in table 2.4 and table 2.5. We can see that Lucy's taste in movies is very close to Eric's (similarity of 0:922) but very different from John's (similarity of 0:938). This means that Eric's ratings can be trusted to predict Lucy's and that Lucy should discard John's opinion on movies or consider the opposite. We also find that the people that like "The Matrix" also like "Die Hard" but hate "Wall-E". Note that these relations were discovered without having any knowledge of the genre, director, or actors of these movies.

The differences in the rating scales of individual users are often more pronounced than the differences in ratings given to individual items. Therefore, while computing the item similarities, it may be more appropriate to compare ratings that are centered on their user mean, instead of their item mean. The Adjusted Cosine (AC) similarity [49], is a modification of the PC item similarity which compares user-mean-centered ratings:

$$AC(i,j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{r}_u)^2 \sum_{u \in \mathcal{U}_{ij}} (r_{uj} - \bar{r}_u)^2}}$$

In some cases, AC similarity has been found to outperform PC similarity on the prediction of ratings using an item-based method [49].

Table 2.4 User-Based Pearson Correlation

	John	Lucy	Eric	Diane
John	1.000	-0.938	-0.839	0.659
Lucy	-0.938	1.000	0.922	-0.787
Eric	-0.839	0.922	1.000	-0.659
Diane	0.659	-0.787	0.659	1.000

Table 2.5 Item-Based Pearson Correlation

	The Matrix	Titanic	Die Hard	Forrest Gump	Wall-E
Matrix	1.000	-0.943	0.882	-0.974	-0.977
Titanic	-0.943	1.000	-0.625	0.931	0.994
Die Hard	0.882	-0.625	1.000	-0.804	-1.000
Forrest Gump	-0.974	0.931	-0.804	1.000	0.930
Wall-E	-0.977	0.994	-1.000	0.930	1.000

2.2.3.2.2 Other Similarity Measures

Several other measures have been proposed to compute similarities between users or items. One of them is the Mean Squared Difference (MSD) [8], which evaluate the similarity between two users u and v as the inverse of the average squared difference between the ratings given by u and v on the same items:

$$MSD(u,v) = \frac{|\mathcal{I}_{uv}|}{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - r_{vi})^2} \quad (2.21)$$

While it could be modified to compute the differences in normalized ratings, the MSD similarity is limited compared to PC similarity because it does not allow to capture negative correlations between user preferences or the appreciation of different items. Having such negative correlations may improve the rating prediction accuracy [59].

2.3 Novelty and Diversity

Novelty can be generally understood as the difference between the present and past experience, whereas diversity relates to the internal differences within parts of an experience. The difference between the two concepts is subtle and close connections can, in fact, be established, depending on the point of view one may take. The general notions of novelty and diversity can be particularized in different ways. For instance, if a music streaming service recommends users a song they have never heard before, they would say this recommendation brings some novelty. Yet if the song is, say, a very canonical music type by some very well-known singer, the involved novelty is considerably less than they would get if the author and style of the music were also original for them. They might also consider that the song is even more novel if, for instance, few of their friends know about it. On the other hand, a music recommendation is diverse if it includes songs of different styles rather than different songs of very similar styles, regardless of whether the songs are original or not for them. Bringing novelty and diversity into play as target properties of the desired outcome means taking a wider perspective on the recommendation problem concerned with final actual recommendation utility, rather than a single quality side such as accuracy [60]. Novelty and diversity are not the only dimensions of recommendation utility one should consider aside from accuracy, but they are fundamental ones. The motivations for enhancing novelty and diversity in recommendations are themselves diverse and can be found in the system, user, and business perspectives. From the system point of view, user actions as implicit evidence of user needs involve a great extent of uncertainty as to what the actual user preferences really are. User clicks and purchases are certainly driven by user interests, but identifying what exactly in an item attracts the user, and generalizing to other items, involves considerable ambiguity. On top of that, system observations are a very limited sample of user activity, whereby recommendation algorithms operate on significantly incomplete knowledge. Furthermore, user interests are complex, highly dynamic, context-dependent, heterogeneous, and even contradictory. Predicting the user needs is, therefore, an inherently difficult task, unavoidably subject to a non-negligible error rate. Diversity can be a good strategy to cope with this uncertainty and optimize the chances that at least some item pleases the user, by widening the range of possible item types and characteristics at which

recommendations aim, rather than bet for a too narrow and risky interpretation of user actions. For instance, a user who has rated the movie “Rango” with the highest value may like it because, in addition to more specific virtues, it is a cartoon, a western, or because it is a comedy. Given the uncertainty about which of the three characteristics may account for the user preference, recommending a movie of each genre generally pays off more than recommending, say three cartoons, as far as three hits do not necessarily bring three times the gain of one-hit e.g. the user might rent just one recommended movie anyway whereas the loss involved in zero hits is considerably worse than achieving a single hit. From this viewpoint, we might say that diversity is not necessarily an opposing goal to accuracy, but in fact, a strategy to optimize the gain drawn from accuracy in matching true user needs in an uncertain environment.

On the other hand, from the user perspective, novelty and diversity are generally a direct source of user satisfaction. Consumer behaviorists have long studied the natural variety-seeking drive in human behavior [19]. The explanation of this drive is commonly divided into direct and derived motivations. The former refers to the inherent satisfaction obtained from “novelty, unexpectedness, change and complexity” [20], and a genuine “desire for the unfamiliar, for alternation among the familiar, and for information” [21], linking to the existence of an ideal level of stimulation, dependent on the individual. Satiation and decreased satisfaction results from the repeated consumption of a product or product characteristic in a decreasing marginal value pattern [61]. As preferences towards discovered products are developed, consumer behavior converges towards a balance between alternating choices and favoring preferred products [62]. Derived motivations include the existence of multiple needs in people, multiple situations, or changes in people’s tastes [20]. Some authors also explain diversity-seeking as a strategy to cope with the uncertainty about one’s own future preference when one will actually consume the choices [63], as e.g. when we choose books and music for a trip. Moreover, novel and diverse recommendations enrich the user experience over time, helping expand the user’s horizon. It is in fact often the case that we approach a recommender system with the explicit intent of discovering something new, developing new interests, and learning. The potential problems of the lack of diversity which may result from too much personalization have recently come to the spotlight with the well-known debate on the so-called filter bubble

[64]. This controversy adds to the motivation for reconciling personalization with a healthy degree of diversity.

Diversity and novelty also find motivation in the underlying businesses in which recommendation technologies are deployed. Customer satisfaction indirectly benefits the business in the form of increased activity, revenues, and customer loyalty. Beyond this, product diversification is a well-known strategy to mitigate risk and expand businesses [64]. Moreover, selling in the long tail is a strategy to draw profit from market niches by selling less of more and getting higher profit margins on cheaper products [65]. All the above general considerations can be of course superseded by particular characteristics of the specific domain, the situation, and the goal of the recommendations, for some of which novelty and diversity are indeed not always needed. For instance, getting a list of similar products (e.g. photo cameras) to the one we are currently inspecting may help us refine our choice among a large set of very similar options. Recommendations can serve as a navigational aid in this type of situation. In other domains, it makes sense to consume the same or very similar items again and again, such as grocery shopping, clothes, etc. The added value of recommendation is probably more limited in such scenarios though, where other kinds of tools may solve our needs (catalog browsers, shopping list assistants, search engines, etc.), and even in these cases we may appreciate some degree of variation in the mix every now and then.

Novelty and diversity are different though related notions, and one finds a rich variety of angles and perspectives on these concepts in the recommender system literature, as well as other fields such as sociology, economy, or ecology. As pointed out at the beginning of this section, novelty generally refers, broadly, to the difference between the present and past experience, whereas diversity relates to the internal differences within parts of an experience. Diversity generally applies to a set of items or “pieces” and has to do with how different the items or pieces are with respect to each other. Variants have been defined by considering different pieces and sets of items. In the basic case, diversity is assessed in the set of items recommended to each user separately (and typically averaged over all users afterward) [13]. But global diversity across sets of items has also been considered, such as the recommendations delivered

to all users [15, 66, 67], recommendations by different systems to the same user [68], or recommendations to a user by the same system over time [69].

The novelty of a set of items can be generally defined as a set function (average, minimum, maximum) on the novelty of the items it contains. We may, therefore, consider novelty as primarily a property of individual items. The novelty of a piece of information generally refers to how different it is with respect to "what has been previously seen or experienced. This is related to novelty in that when a set is diverse, each item is "novel" with respect to the rest of the set. Moreover, a system that promotes novel results tends to generate global diversity over time in the user experience; and also enhances the global "diversity of sales" from the system perspective. Multiple variants of novelty arise by considering the fact that novelty is relative to a context of experience, as we shall discuss.

Different nuances have been considered in the concept of novelty. A simple definition of novelty can consist of the (binary) absence of an item in the context of reference (prior experience). We may use adjectives such as unknown or unseen for this notion of identity-based novelty [18]. Long-tail notions of novelty are elaborations of this concept, as they are defined in terms of the number of users who would specifically know an item [16, 66, 70]. But we may also consider how different or similar an unseen item is with respect to known items, generally—but not necessarily—on a graded scale. Adjectives such as unexpected, surprising and unfamiliar have been used to refer to this variant of novelty. Unfamiliarity and identity novelty can be related by trivially defining similarity as equality, i.e. two items are "similar" if and only if they are the same item. Finally, the notion of serendipity is used to mean novelty plus a positive emotional response, in other words, an item is serendipitous if it is novel, unknown or unfamiliar and relevant [71, 72].

The study about the diversity and novelty involved in recommendations can be in several aspects such as the diversity (in tastes, behavior, demographics, etc.) of the end-user population, or the product stock, the sellers, or in general the environment in which recommenders operate. While some works in the field have addressed the diversity in user behavior [73, 74], we will mostly focus on those aspects a recommender system has a direct hold on, namely the properties of its own output.

2.4 Relevant Research about Similarity Diversity Challenge

While this study objective is to investigate the relation between the diversity of recommended items and the accuracy of prediction of the recommender system, there are many pieces of research that also have the similar goal to investigate novelty and diversity in recommender systems that can be used as the references in this study. G. Adomavicius and Y. Kwon [75] stated that collaborative filtering and, more generally, recommender systems represent an increasingly popular and important set of personalization technologies that help people navigate through the vast amounts of information. The performance of recommender systems can be evaluated along several dimensions, such as the accuracy of recommendations for each user and the diversity of recommendations across different users. Intuitively, there is a tradeoff between accuracy and diversity, because high accuracy may often be obtained by safely recommending to users the most popular ("bestselling") items, which can lead to the reduction in recommendation diversity, i.e., less personalized recommendations. And conversely, higher diversity can be achieved by trying to uncover and recommend highly idiosyncratic/personalized items for each user, which are inherently more difficult to predict and, thus, may lead to a decrease in recommendation accuracy. In their research, they explored different ways to overcome this accuracy-diversity tradeoff, and in this research, we discuss a variance-based approach that can improve both the accuracy and diversity of recommendations obtained from a traditional collaborative filtering technique then provided empirical results based on several real-world movie rating datasets. This research adopted a variance-based approach to improving the accuracy and diversity of recommendations using a traditional CF technique and empirically demonstrated how this new approach can overcome the accuracy-diversity tradeoff.

A. Said et al. [76] stated that one of the current challenges concerning improving recommender systems consists of finding ways of increasing serendipity and diversity, without compromising the precision and recall of the system. One possible way to approach this problem is to complement a standard recommender by another recommender "orthogonal" to the standard one, i.e. one that recommends different items than the standard. In their study, an investigation was done to find out that an inverted nearest-neighbor model, k-furthest neighbor, was suitable for complementing

a traditional k-NN recommender or not?. They compare the recommendations obtained when recommending items disliked by people least similar to oneself to those obtained by recommending items liked by those most similar to oneself. In their other research [77], this topic was investigated further. They stated that collaborative filtering recommender systems often use nearest neighbor methods to identify candidate items. In their study, an inverted neighborhood model, k-Furthest Neighbors, was presented to identify less ordinary neighbor-hoods for the purpose of creating more diverse recommendations. The approach was evaluated two-fold, once in a traditional information retrieval evaluation setting where the model is trained and validated on a split train/test set, and once through an online user study to identify users' perceived quality of the recommender. A standard k-nearest neighbor recommendation was used as a baseline in both evaluation settings.

Chapter 3
Methodology

In this research, the process will start from understanding data in order to design the plan for study and analysis. After that the system will be designed and developed in order to make analyses. Firstly, data exploration will be conducted in order to understand the dataset. Secondly, data wrangling will be conducted to make dataset in the form that can be used with model. Lastly, experiment need to be designed and conducted in order to analyze the diversity and accuracy in the system

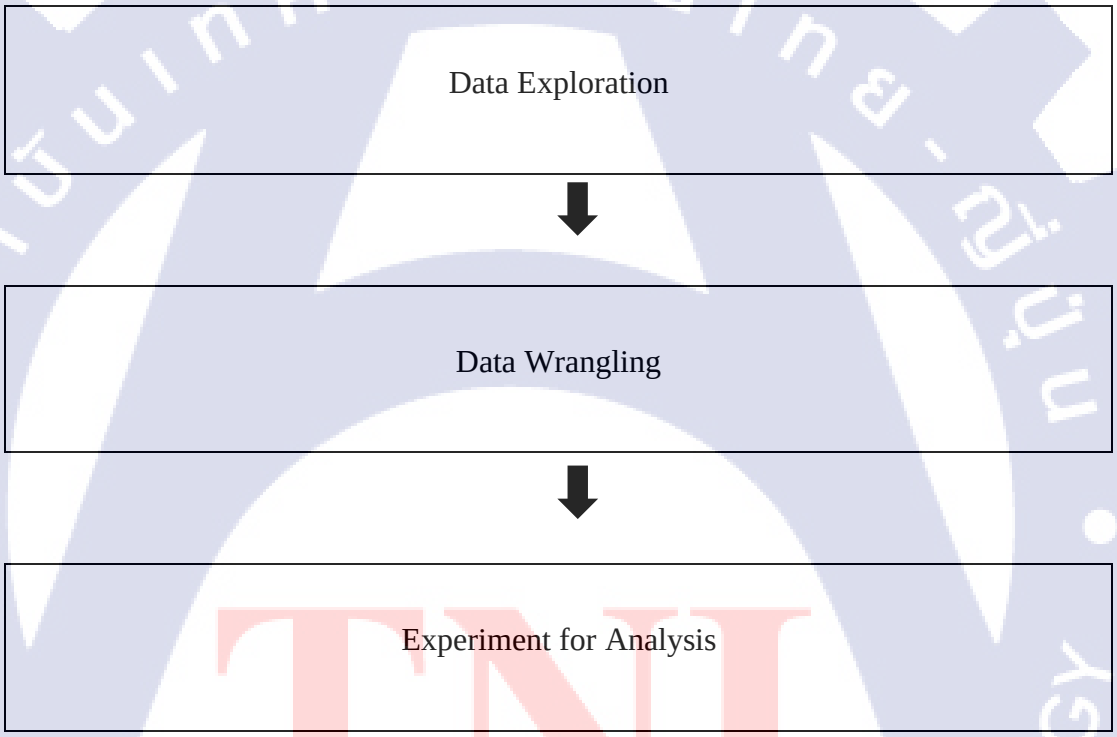


Figure 3.1 study process

3.1 Data Exploration

Undergraduates' academic data was received from the Muroran Institute of Technology (MIT) Information Technology (IT) department. It was the data between the academic year of Heisei 18 (2006) to Heisei 20 (2008). The reason this period of data was used is that in the academic year of Heisei 18, MIT started a new set of courses

for undergraduates and also in the academic year of Heisei 21 (2009), MIT separated undergraduates' courses of IT departments into 4 departments. These reasons caused the received data to become too diverse so in this study, only data from the range between the academic year of Heisei 18 and the academic year of Heisei 20; Heisei 18, Heisei 19, and Heisei 20; was used for system development and analysis.

Starting with data exploration, the first thing to be done is to explore the data in order to understand it. It was found that the subjects that undergraduates need to enroll was separated into several types of course but can approximately separate into 2 main types as 必修 (compulsory subjects for enrolled) and 選択 (select subjects for enrolled). Because all undergraduates need to enroll in the compulsory courses so it is not necessary to include them as the recommended item cause every undergrad needs to enroll it anyway so the select subject was only focused on.

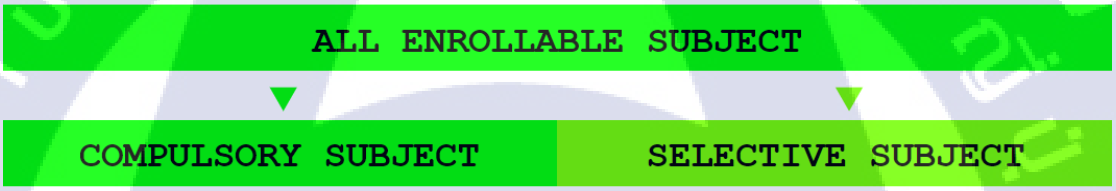


Figure 3.2 MIT enrollable subjects

As was discussed earlier, the only select subject will be used in this study. Table 3.2, 3.3, and 3.4 that are shown below are the list of the select subjects in each academic year.

Table 3.1 select subjects in the academic year of Heisei 18

Course's Type	Course's Name
教育 主専門 主共通 選択	物理学Ａ
教育 主専門 主共通 選択	物理学Ｂ

Table 3.1 select subjects in the academic year of Heisei 18 (Continued)

Course's Type	Course's Name
教育 主専門 主共通 選択	物理学Ｃ
教育 主専門 主共通 選択	物理学実験
教育 主専門 主共通 選択	基礎化学
教育 主専門 主共通 選択	化学実験
教育 主専門 主共通 選択	図学I
教育 主専門 主共通 選択	図学II
教育 主専門 主学科 選択	学外実習
教育 主専門 主学科 選択	数値解析
教育 主専門 主学科 選択	情報理論
教育 主専門 主学科 選択	人工知能
教育 主専門 主学科 選択	ディジタル信号処理
教育 主専門 主学科 選択	ファイルとデータベース
教育 主専門 主学科 選択	システム工学
教育 主専門 主学科 選択	データの統計解析
教育 主専門 主学科 選択	視覚情報処理
教育 主専門 主学科 選択	認識と学習
教育 主専門 主学科 選択	マルチメディア工学

Table 3.1 select subjects in the academic year of Heisei 18 (Continued)

Course's Type	Course's Name
教育 主専門 主学科 選択	システム制御理論
教育 主専門 主学科 選択	情報関連法規
教育 主専門 主学科 選択	情報と職業
教育 主専門 主学科 選択	電子情報回路
教育 主専門 主学科 選択	プログラミングＢ
教育 主専門 主学科 選択	情報通信工学
教育 主専門 主学科 選択	データ統計解析応用演習
教育 主専門 主学科 選択	認識と学習応用演習
教育 主専門 主学科 選択	組込みシステム
教育 主専門 主学科 選択	人工知能応用演習
教育 主専門 主学科 選択	プログラミングＢ応用演習
教育 主専門 主学科 選択	視覚情報処理応用演習
教育 主専門 主学科 選択	言語処理系論
教育 主専門 主学科 選択	研究課題調査
教育 主専門 主学科 選択	システム制御工学
教育 副専門 副共通 選択	日本の憲法
教育 副専門 副共通 選択	現代の社会Ａ

Table 3.1 select subjects in the academic year of Heisei 18 (Continued)

Course's Type	Course's Name
教育 副専門 副共通 選択	こころの科学
教育 副専門 副共通 選択	哲学入門Ａ
教育 副専門 副共通 選択	哲学入門Ｂ
教育 副専門 副共通 選択	経済のしくみＡ
教育 副専門 副共通 選択	人間と文化
教育 副専門 副共通 選択	経済のしくみＢ
教育 副専門 副共通 選択	日本の歴史
教育 副専門 副共通 選択	現代の社会Ｂ
教育 副専門 副共通 選択	西洋の歴史
教育 副専門 副共通 選択	インター・サイエンスＡ（建設）
教育 副専門 副共通 選択	インター・サイエンスＢ（機械）
教育 副専門 副共通 選択	インター・サイエンスＤ（電電）
教育 副専門 副共通 選択	インター・サイエンスＥ（材物）
教育 副専門 副共通 選択	インター・サイエンスＦ（応化）
教育 副専門 副共通 選択	数学入門
教育 副専門 副共通 選択	生物学入門
教育 副専門 副共通 選択	環境科学入門

Table 3.1 select subjects in the academic year of Heisei 18 (Continued)

Course's Type	Course's Name
教育 副専門 副共通 選択	現代工学の課題
教育 副専門 副共通 選択	地球科学入門
教育 副専門 副共通 選択	ＴＯＥＩＣ英語演習
教育 副専門 副共通 選択	英語コミュニケーション演習I
教育 副専門 副共通 選択	英語コミュニケーション演習II
教育 副専門 副共通 選択	ＴＯＥＦＬ英語演習
教育 副専門 副共通 選択	応用英語演習
教育 副専門 副共通 選択	ドイツ語Ⅰａ
教育 副専門 副共通 選択	ロシア語Ⅰａ
教育 副専門 副共通 選択	中国語Ⅰａ
教育 副専門 副共通 選択	ドイツ語Ⅰｂ
教育 副専門 副共通 選択	ロシア語Ⅰｂ
教育 副専門 副共通 選択	中国語Ⅰｂ
教育 副専門 副共通 選択	ドイツ語II
教育 副専門 副共通 選択	ロシア語II
教育 副専門 副共通 選択	中国語II
教育 副専門 副共通 選択	スポーツ実習 a

Table 3.1 select subjects in the academic year of Heisei 18 (Continued)

Course's Type	Course's Name
教育 副専門 副共通 選択	スポーツ実習ｂ
教育 副専門 副共通 選択	スポーツ実習ｃ
教育 副専門 副共通 選択	スポーツ実習ｄ
教育 副専門 副共通 選択	異文化交流Ｂ
教育 副専門 副共通 選択	キャリア・デザイン
教育 副専門 副共通 選択	文学創作演習
教育 副専門 副コース 選択	経済事情
教育 副専門 副コース 選択	社会環境基礎論
教育 副専門 副コース 選択	基層文化論
教育 副専門 副コース 選択	環境経済論
教育 副専門 副コース 選択	環境法制
教育 副専門 副コース 選択	社会環境論
教育 副専門 副コース 選択	社会環境アセスメント論
教育 副専門 副コース 選択	ゼミナール「環境と社会」
教育 副専門 副コース 選択	環境生物学
教育 副専門 副コース 選択	生活環境科学
教育 副専門 副コース 選択	生態保全論

Table 3.1 select subjects in the academic year of Heisei 18 (Continued)

Course's Type	Course's Name
教育 副専門 副コース 選択	環境有機化学
教育 副専門 副コース 選択	地球環境化学
教育 副専門 副コース 選択	自然再生論
教育 副専門 副コース 選択	現代民主主義論
教育 副専門 副コース 選択	ヨーロッパ史
教育 副専門 副コース 選択	日本近現代史Ａ
教育 副専門 副コース 選択	平和と憲法
教育 副専門 副コース 選択	基本的人権論
教育 副専門 副コース 選択	現代自由論
教育 副専門 副コース 選択	国際関係論
教育 副専門 副コース 選択	日本近現代史Ｂ
教育 副専門 副コース 選択	ゼミナール「市民と公共」Ａ
教育 副専門 副コース 選択	医の科学Ａ
教育 副専門 副コース 選択	機能回復の生理学
教育 副専門 副コース 選択	環境と資源
教育 副専門 副コース 選択	地球科学
教育 副専門 副コース 選択	医の科学Ｂ

Table 3.1 select subjects in the academic year of Heisei 18 (Continued)

Course's Type	Course's Name
教育 副専門 副コース 選択	ゼミナール「市民と公共」B
教育 副専門 副コース 選択	外国文学
教育 副専門 副コース 選択	博物館学
教育 副専門 副コース 選択	メンタルヘルス論
教育 副専門 副コース 選択	現代心理学
教育 副専門 副コース 選択	日本文学
教育 副専門 副コース 選択	アジアの文化
教育 副専門 副コース 選択	人間と文学
教育 副専門 副コース 選択	青少年と文化
教育 副専門 副コース 選択	ヨーロッパの文化
教育 副専門 副コース 選択	ゼミナール「人間と文化」
教育 副専門 副コース 選択	からだの科学
教育 副専門 副コース 選択	行動の科学
教育 副専門 副コース 選択	感性の科学
教育 副専門 副コース 選択	人間の環境化学
教育 副専門 副コース 選択	水圏生物科学
教育 副専門 副コース 選択	認識の哲学

Table 3.1 select subjects in the academic year of Heisei 18 (Continued)

Course's Type	Course's Name
教育 副専門 副コース 選択	認知科学論
教育 副専門 副コース 選択	言語の哲学
教育 副専門 副コース 選択	科学と倫理
教育 副専門 副コース 選択	現代論理学
教育 副専門 副コース 選択	自己理解のサイエンス
教育 副専門 副コース 選択	認知科学の諸問題
教育 副専門 副コース 選択	ゼミナール「思考と数理」A
教育 副専門 副コース 選択	距離空間
教育 副専門 副コース 選択	線形空間
教育 副専門 副コース 選択	代数学概論
教育 副専門 副コース 選択	解析学概論
教育 副専門 副コース 選択	数学考究
教育 副専門 副コース 選択	ゼミナール「思考と数理」B
教育 副専門 日本語	日本語 A-1
教育 副専門 日本語	日本語 A-2
教育 副専門 日本語	日本語 B-1
教育 副専門 日本語	日本語 B-2

Table 3.1 select subjects in the academic year of Heisei 18 (Continued)

Course's Type	Course's Name
教育 副専門 日本語	日本語 C-1
教育 副専門 日本語	日本語 C-2
教職 教職科目	教職原論
教職 教職科目	教育学概論
教職 教職科目	教育心理学
教職 教職科目	対人関係論
教職 教職科目	教育内容論
教職 教職科目	情報教育法
教職 教職科目	教育方法論
教職 教職科目	教育工学
教職 教職科目	進路指導
教職 教職科目	教育相談
教職 教職科目	総合演習
教職 教職科目	教育実習

Table 3.2 select subjects in the academic year of Heisei 19

Course's Type	Course's Name
教育 主専門 主共通 選択	物理学Ａ
教育 主専門 主共通 選択	物理学Ｂ
教育 主専門 主共通 選択	物理学Ｃ
教育 主専門 主共通 選択	物理学実験
教育 主専門 主共通 選択	基礎化学
教育 主専門 主共通 選択	化学実験
教育 主専門 主共通 選択	図学Ⅰ
教育 主専門 主共通 選択	図学Ⅱ
教育 主専門 主学科 選択	学外実習
教育 主専門 主学科 選択	数値解析
教育 主専門 主学科 選択	情報理論
教育 主専門 主学科 選択	情報計測工学
教育 主専門 主学科 選択	人工知能
教育 主専門 主学科 選択	ディジタル信号処理
教育 主専門 主学科 選択	ファイルとデータベース
教育 主専門 主学科 選択	システム工学
教育 主専門 主学科 選択	データの統計解析
教育 主専門 主学科 選択	視覚情報処理

Table 3.2 select subjects in the academic year of Heisei 19 (Continued)

Course's Type	Course's Name
教育 主専門 主学科 選択	認識と学習
教育 主専門 主学科 選択	マルチメディア工学
教育 主専門 主学科 選択	システム制御理論
教育 主専門 主学科 選択	情報関連法規
教育 主専門 主学科 選択	情報と職業
教育 主専門 主学科 選択	電子情報回路
教育 主専門 主学科 選択	プログラミングＢ
教育 主専門 主学科 選択	情報通信工学
教育 主専門 主学科 選択	データ統計解析応用演習
教育 主専門 主学科 選択	認識と学習応用演習
教育 主専門 主学科 選択	組込みシステム
教育 主専門 主学科 選択	人工知能応用演習
教育 主専門 主学科 選択	プログラミングＢ応用演習
教育 主専門 主学科 選択	視覚情報処理応用演習
教育 主専門 主学科 選択	言語処理系論
教育 主専門 主学科 選択	研究課題調査
教育 副専門 副共通 選択	日本の憲法
教育 副専門 副共通 選択	現代の社会Ａ

Table 3.2 select subjects in the academic year of Heisei 19 (Continued)

Course's Type	Course's Name
教育 副専門 副共通 選択	こころの科学
教育 副専門 副共通 選択	哲学入門Ａ
教育 副専門 副共通 選択	哲学入門Ｂ
教育 副専門 副共通 選択	経済のしくみＡ
教育 副専門 副共通 選択	人間と文化
教育 副専門 副共通 選択	経済のしくみＢ
教育 副専門 副共通 選択	日本の歴史
教育 副専門 副共通 選択	現代の社会Ｂ
教育 副専門 副共通 選択	西洋の歴史
教育 副専門 副共通 選択	インター・サイエンスＡ（建設）
教育 副専門 副共通 選択	インター・サイエンスＢ（機械）
教育 副専門 副共通 選択	インター・サイエンスＤ（電電）
教育 副専門 副共通 選択	インター・サイエンスＥ（材物）
教育 副専門 副共通 選択	インター・サイエンスＦ（応化）
教育 副専門 副共通 選択	数学入門
教育 副専門 副共通 選択	生物学入門
教育 副専門 副共通 選択	環境科学入門
教育 副専門 副共通 選択	現代工学の課題

Table 3.2 select subjects in the academic year of Heisei 19 (Continued)

Course's Type	Course's Name
教育 副専門 副共通 選択	地球科学入門
教育 副専門 副共通 選択	ＴＯＥＩＣ英語演習
教育 副専門 副共通 選択	英語コミュニケーション演習Ⅰ
教育 副専門 副共通 選択	英語コミュニケーション演習Ⅱ
教育 副専門 副共通 選択	ＴＯＥＦＬ英語演習
教育 副専門 副共通 選択	応用英語演習
教育 副専門 副共通 選択	ドイツ語Ⅰａ
教育 副専門 副共通 選択	ロシア語Ⅰａ
教育 副専門 副共通 選択	中国語Ⅰａ
教育 副専門 副共通 選択	ドイツ語Ⅰｂ
教育 副専門 副共通 選択	ロシア語Ⅰｂ
教育 副専門 副共通 選択	中国語Ⅰｂ
教育 副専門 副共通 選択	ドイツ語Ⅱ
教育 副専門 副共通 選択	ロシア語Ⅱ
教育 副専門 副共通 選択	中国語Ⅱ
教育 副専門 副共通 選択	スポーツ実習ａ
教育 副専門 副共通 選択	スポーツ実習ｂ
教育 副専門 副共通 選択	スポーツ実習ｃ

Table 3.2 select subjects in the academic year of Heisei 19 (Continued)

Course's Type	Course's Name
教育 副専門 副共通 選択	スポーツ実習ｄ
教育 副専門 副共通 選択	異文化交流Ｂ
教育 副専門 副共通 選択	キャリア・デザイン
教育 副専門 副共通 選択	文学創作演習
教育 副専門 副共通 選択	海外語学研修
教育 副専門 副共通 選択	地域再生システム論
教育 副専門 副コース 選択	経済事情
教育 副専門 副コース 選択	社会環境基礎論
教育 副専門 副コース 選択	基層文化論
教育 副専門 副コース 選択	環境経済論
教育 副専門 副コース 選択	環境法制
教育 副専門 副コース 選択	社会環境論
教育 副専門 副コース 選択	社会環境アセスメント論
教育 副専門 副コース 選択	ゼミナール「環境と社会」
教育 副専門 副コース 選択	環境生物学
教育 副専門 副コース 選択	生活環境科学
教育 副専門 副コース 選択	生態保全論
教育 副専門 副コース 選択	環境有機化学

Table 3.2 select subjects in the academic year of Heisei 19 (Continued)

Course's Type	Course's Name
教育 副専門 副コース 選択	地球環境化学
教育 副専門 副コース 選択	自然再生論
教育 副専門 副コース 選択	現代民主主義論
教育 副専門 副コース 選択	ヨーロッパ史
教育 副専門 副コース 選択	日本近現代史Ａ
教育 副専門 副コース 選択	平和と憲法
教育 副専門 副コース 選択	基本的人権論
教育 副専門 副コース 選択	現代自由論
教育 副専門 副コース 選択	国際関係論
教育 副専門 副コース 選択	日本近現代史Ｂ
教育 副専門 副コース 選択	ゼミナール「市民と公共」Ａ
教育 副専門 副コース 選択	医の科学Ａ
教育 副専門 副コース 選択	機能回復の生理学
教育 副専門 副コース 選択	環境と資源
教育 副専門 副コース 選択	地球科学
教育 副専門 副コース 選択	医の科学Ｂ
教育 副専門 副コース 選択	外国文学
教育 副専門 副コース 選択	博物館学

Table 3.2 select subjects in the academic year of Heisei 19 (Continued)

Course's Type	Course's Name
教育 副専門 副コース 選択	メンタルヘルス論
教育 副専門 副コース 選択	現代心理学
教育 副専門 副コース 選択	日本文学
教育 副専門 副コース 選択	アジアの文化
教育 副専門 副コース 選択	人間と文学
教育 副専門 副コース 選択	青少年と文化
教育 副専門 副コース 選択	ヨーロッパの文化
教育 副専門 副コース 選択	ゼミナール「人間と文化」
教育 副専門 副コース 選択	からだの科学
教育 副専門 副コース 選択	行動の科学
教育 副専門 副コース 選択	感性の科学
教育 副専門 副コース 選択	人間の環境化学
教育 副専門 副コース 選択	水圏生物科学
教育 副専門 副コース 選択	認識の哲学
教育 副専門 副コース 選択	認知科学論
教育 副専門 副コース 選択	言語の哲学
教育 副専門 副コース 選択	科学と倫理
教育 副専門 副コース 選択	現代論理学

Table 3.2 select subjects in the academic year of Heisei 19 (Continued)

Course's Type	Course's Name
教育 副専門 副コース 選択	自己理解のサイエンス
教育 副専門 副コース 選択	認知科学の諸問題
教育 副専門 副コース 選択	ゼミナール「思考と数理」A
教育 副専門 副コース 選択	距離空間
教育 副専門 副コース 選択	線形空間
教育 副専門 副コース 選択	代数学概論
教育 副専門 副コース 選択	解析学概論
教育 副専門 副コース 選択	数学考究
教育 副専門 日本語	日本語 A-1
教育 副専門 日本語	日本語 A-2
教育 副専門 日本語	日本語 B-1
教育 副専門 日本語	日本語 B-2
教育 副専門 日本語	日本語 C-1
教育 副専門 日本語	日本語 D-1
教職 教職科目	教職原論
教職 教職科目	教育学概論
教職 教職科目	教育心理学
教職 教職科目	対人関係論

Table 3.2 select subjects in the academic year of Heisei 19 (Continued)

Course's Type	Course's Name
教職 教職科目	教育内容論
教職 教職科目	情報教育法
教職 教職科目	教育方法論
教職 教職科目	教育工学
教職 教職科目	進路指導
教職 教職科目	教育相談
教職 教職科目	総合演習
教職 教職科目	教育実習

Table 3.3 select subjects in the academic year of Heisei 20

Course's Type	Course's Name
教育 主専門 主共通 選択	物理学Ａ
教育 主専門 主共通 選択	物理学Ｂ
教育 主専門 主共通 選択	物理学Ｃ
教育 主専門 主共通 選択	物理学実験
教育 主専門 主共通 選択	基礎化学
教育 主専門 主共通 選択	化学実験
教育 主専門 主共通 選択	図学Ⅰ
教育 主専門 主共通 選択	図学Ⅱ

Table 3.3 select subjects in the academic year of Heisei 20 (Continued)

Course's Type	Course's Name
教育 主専門 主学科 選択	学外実習
教育 主専門 主学科 選択	数値解析
教育 主専門 主学科 選択	情報理論
教育 主専門 主学科 選択	情報計測工学
教育 主専門 主学科 選択	人工知能
教育 主専門 主学科 選択	ディジタル信号処理
教育 主専門 主学科 選択	ファイルとデータベース
教育 主専門 主学科 選択	システム工学
教育 主専門 主学科 選択	視覚情報処理
教育 主専門 主学科 選択	認識と学習
教育 主専門 主学科 選択	マルチメディア工学
教育 主専門 主学科 選択	システム制御理論
教育 主専門 主学科 選択	情報関連法規
教育 主専門 主学科 選択	情報と職業
教育 主専門 主学科 選択	電子情報回路
教育 主専門 主学科 選択	プログラミングＢ
教育 主専門 主学科 選択	情報通信工学
教育 主専門 主学科 選択	認識と学習応用演習

Table 3.3 select subjects in the academic year of Heisei 20 (Continued)

Course's Type	Course's Name
教育 主専門 主学科 選択	組込みシステム
教育 主専門 主学科 選択	人工知能応用演習
教育 主専門 主学科 選択	プログラミングＢ応用演習
教育 主専門 主学科 選択	視覚情報処理応用演習
教育 主専門 主学科 選択	言語処理系論
教育 主専門 主学科 選択	研究課題調査
教育 主専門 主学科 選択	確率・統計応用演習
教育 副専門 副共通 選択	日本の憲法
教育 副専門 副共通 選択	現代の社会Ａ
教育 副専門 副共通 選択	こころの科学
教育 副専門 副共通 選択	哲学入門Ａ
教育 副専門 副共通 選択	哲学入門Ｂ
教育 副専門 副共通 選択	経済のしくみＡ
教育 副専門 副共通 選択	人間と文化
教育 副専門 副共通 選択	経済のしくみＢ
教育 副専門 副共通 選択	日本の歴史
教育 副専門 副共通 選択	現代の社会Ｂ
教育 副専門 副共通 選択	西洋の歴史

Table 3.3 select subjects in the academic year of Heisei 20 (Continued)

Course's Type	Course's Name
教育 副専門 副共通 選択	インター・サイエンスＡ（建設）
教育 副専門 副共通 選択	インター・サイエンスＢ（機械）
教育 副専門 副共通 選択	インター・サイエンスＤ（電電）
教育 副専門 副共通 選択	インター・サイエンスＥ（材物）
教育 副専門 副共通 選択	インター・サイエンスＦ（応化）
教育 副専門 副共通 選択	数学入門
教育 副専門 副共通 選択	生物学入門
教育 副専門 副共通 選択	環境科学入門
教育 副専門 副共通 選択	現代工学の課題
教育 副専門 副共通 選択	地球科学入門
教育 副専門 副共通 選択	ＴＯＥＩＣ英語演習
教育 副専門 副共通 選択	英語コミュニケーション演習Ⅰ
教育 副専門 副共通 選択	英語コミュニケーション演習Ⅱ
教育 副専門 副共通 選択	ＴＯＥＦＬ英語演習
教育 副専門 副共通 選択	応用英語演習
教育 副専門 副共通 選択	ドイツ語Ⅰａ
教育 副専門 副共通 選択	ロシア語Ⅰａ
教育 副専門 副共通 選択	中国語Ⅰａ

Table 3.3 select subjects in the academic year of Heisei 20 (Continued)

Course's Type	Course's Name
教育 副専門 副共通 選択	ドイツ語Ⅰｂ
教育 副専門 副共通 選択	ロシア語Ⅰｂ
教育 副専門 副共通 選択	中国語Ⅰｂ
教育 副専門 副共通 選択	ドイツ語Ⅱ
教育 副専門 副共通 選択	ロシア語Ⅱ
教育 副専門 副共通 選択	中国語Ⅱ
教育 副専門 副共通 選択	スポーツ実習ａ
教育 副専門 副共通 選択	スポーツ実習ｂ
教育 副専門 副共通 選択	スポーツ実習ｃ
教育 副専門 副共通 選択	スポーツ実習ｄ
教育 副専門 副共通 選択	異文化交流Ａ
教育 副専門 副共通 選択	異文化交流Ｂ
教育 副専門 副共通 選択	キャリア・デザイン
教育 副専門 副共通 選択	文学創作演習
教育 副専門 副共通 選択	社会体験実習
教育 副専門 副共通 選択	海外語学研修
教育 副専門 副共通 選択	海外研修
教育 副専門 副コース 選択	経済事情

Table 3.3 select subjects in the academic year of Heisei 20 (Continued)

Course's Type	Course's Name
教育 副専門 副コース 選択	社会環境基礎論
教育 副専門 副コース 選択	基層文化論
教育 副専門 副コース 選択	環境経済論
教育 副専門 副コース 選択	環境法制
教育 副専門 副コース 選択	社会環境論
教育 副専門 副コース 選択	社会環境アセスメント論
教育 副専門 副コース 選択	環境生物学
教育 副専門 副コース 選択	生活環境科学
教育 副専門 副コース 選択	生態保全論
教育 副専門 副コース 選択	環境有機化学
教育 副専門 副コース 選択	地球環境化学
教育 副専門 副コース 選択	自然再生論
教育 副専門 副コース 選択	現代民主主義論
教育 副専門 副コース 選択	ヨーロッパ史
教育 副専門 副コース 選択	日本近現代史Ａ
教育 副専門 副コース 選択	平和と憲法
教育 副専門 副コース 選択	基本的人権論
教育 副専門 副コース 選択	国際関係論

Table 3.3 select subjects in the academic year of Heisei 20 (Continued)

Course's Type	Course's Name
教育 副専門 副コース 選択	日本近現代史Ｂ
教育 副専門 副コース 選択	医の科学Ａ
教育 副専門 副コース 選択	機能回復の生理学
教育 副専門 副コース 選択	環境と資源
教育 副専門 副コース 選択	地球科学
教育 副専門 副コース 選択	医の科学Ｂ
教育 副専門 副コース 選択	外国文学
教育 副専門 副コース 選択	博物館学
教育 副専門 副コース 選択	メンタルヘルス論
教育 副専門 副コース 選択	現代心理学
教育 副専門 副コース 選択	日本文学
教育 副専門 副コース 選択	アジアの文化
教育 副専門 副コース 選択	人間と文学
教育 副専門 副コース 選択	青少年と文化
教育 副専門 副コース 選択	ヨーロッパの文化
教育 副専門 副コース 選択	ゼミナール「人間と文化」
教育 副専門 副コース 選択	からだの科学
教育 副専門 副コース 選択	行動の科学

Table 3.3 select subjects in the academic year of Heisei 20 (Continued)

Course's Type	Course's Name
教育 副専門 副コース 選択	感性の科学
教育 副専門 副コース 選択	人間の環境化学
教育 副専門 副コース 選択	水圏生物科学
教育 副専門 副コース 選択	認識の哲学
教育 副専門 副コース 選択	認知科学論
教育 副専門 副コース 選択	言語の哲学
教育 副専門 副コース 選択	科学と倫理
教育 副専門 副コース 選択	現代論理学
教育 副専門 副コース 選択	自己理解のサイエンス
教育 副専門 副コース 選択	認知科学の諸問題
教育 副専門 副コース 選択	ゼミナール「思考と数理」A
教育 副専門 副コース 選択	距離空間
教育 副専門 副コース 選択	線形空間
教育 副専門 副コース 選択	代数学概論
教育 副専門 副コース 選択	解析学概論
教育 副専門 副コース 選択	数学考究
教育 副専門 副コース 選択	ゼミナール「思考と数理」B
教育 副専門 副コース 選択	地方自治論

Table 3.3 select subjects in the academic year of Heisei 20 (Continued)

Course's Type	Course's Name
教育 副専門 日本語	日本語 B-1
教育 副専門 日本語	日本語 C-1
教職 教職科目	教職原論
教職 教職科目	教育学概論
教職 教職科目	教育心理学
教職 教職科目	対人関係論
教職 教職科目	教育内容論
教職 教職科目	情報教育法
教職 教職科目	教育方法論
教職 教職科目	教育工学
教職 教職科目	進路指導
教職 教職科目	教育相談
教職 教職科目	総合演習
教職 教職科目	教育実習

It can be seen that select subject in each academic are slightly different, comparing subjects in the same group. This created some diverse between selective choices for undergraduates.

Also, it was found that in the data, there were some Japanese characters used to replace the score value of subjects as the following:

1. 欠 can be translated as “all absent”, this was decided to replace it with the numerical value equal to 0

2. 認 can be translated as “pass”, this was decided to replace it with the numerical value equal to 60

3. M can be translated as “withdraw”, this was decided to replace it with the numerical value equal to 0

4. 不 can be translated as “not pass”, this was decided to replace it with the numerical value as 0

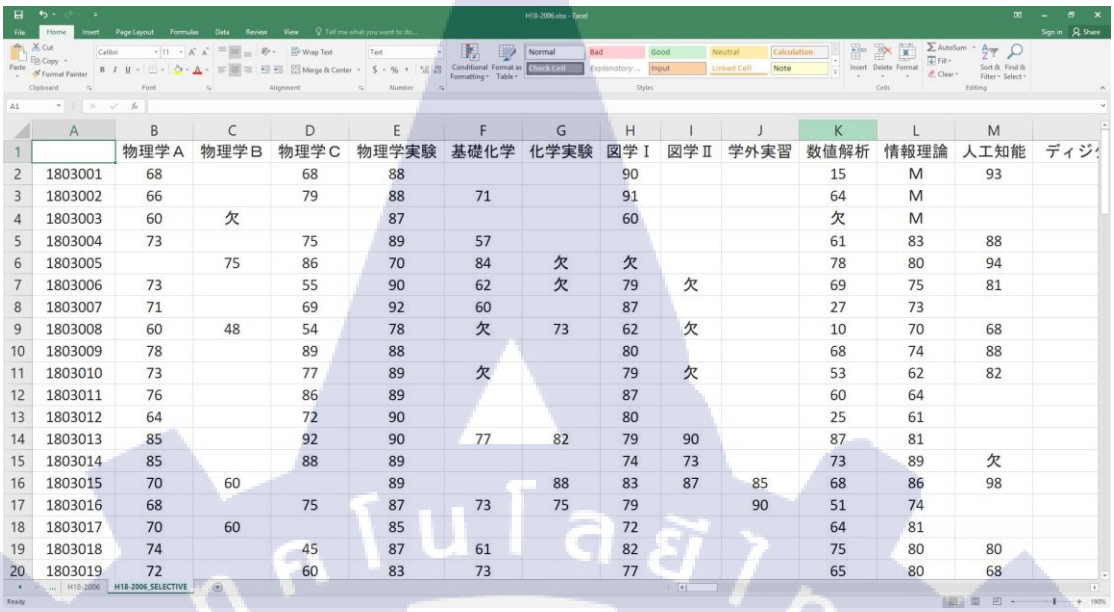
5. F can be translated as “fail”, this was decided to replace it with the numerical value equal to 0

Table 3.4 data cleansing description

CHARACTER	MEANING	VALUE
欠	“all absent”	0
認	“pass”	60
M	“with draw”	0
不	“not pass”	0
F	“fail”	0

3.2 Data Wrangling

After the exploration of data, data wrangling was proceeded to make the data in the form that can be used. Firstly, the data was originally in the form of .xlsx file and have the form as in Figure 3.3 The original form of these files cannot be used because, in this form, the subject that undergraduate did not enroll will be treated as missing data so we need to transform this data into a form that can be used.



	A	B	C	D	E	F	G	H	I	J	K	L	M	
1		物理学A	物理学B	物理学C	物理学実験	基礎化学	化学実験	図学Ⅰ	図学Ⅱ	学外実習	数値解析	情報理論	人工知能	ディジ
2	1803001	68		68	88			90			15	M	93	
3	1803002	66		79	88	71		91			64	M		
4	1803003	60	欠		87			60			欠	M		
5	1803004	73		75	89	57					61	83	88	
6	1803005		75	86	70	84	欠	欠			78	80	94	
7	1803006	73		55	90	62	欠	79	欠		69	75	81	
8	1803007	71		69	92	60		87			27	73		
9	1803008	60	48	54	78	欠	73	62	欠		10	70	68	
10	1803009	78		89	88			80			68	74	88	
11	1803010	73		77	89	欠		79	欠		53	62	82	
12	1803011	76		86	89			87			60	64		
13	1803012	64		72	90			80			25	61		
14	1803013	85		92	90	77	82	79	90		87	81		
15	1803014	85		88	89			74	73		73	89	欠	
16	1803015	70	60		89		88	83	87	85	68	86	98	
17	1803016	68		75	87	73	75	79		90	51	74		
18	1803017	70	60		85			72			64	81		
19	1803018	74		45	87	61		82			75	80	80	
20	1803019	72		60	83	73		77			65	80	68	

Figure 3.3 original form of data

We use Python as the programing language to deal with the data because Python has many packages that contain coding and command that can be used easily for developing recommender system. Google Collaboratory was used as the platform for coding because it is easier to code from different places because it can access file store online in sources like Google Drive, we don't need to save file into the device we coded on and it can use server from google to run and can increase its performance by setting. Its advantage was its runtime that cannot stand long computing time for the vast amount of data since our data was not that vast, we can use for no problem. Firstly, we need to set Google Drive as the directory for Google Collaboratory in order to make the stored files in Google Drive can be accessed. The data transformation process will start by replacing unique character in data with numeric score that can be used in model and also remove Nan missing value from the dataset. Following by transforming data in original form to the form that can be used in models.

ID	解析 A	解析 C	...
1803001	68	68	...
...	...	...	...

ID	SUBJECT	SCORE
1803001	解析 A	68
1803001	解析 C	68
...	...	...

Figure 3.4 User-Based proper form of data

ID	解析 A	...
1803001	68	...
1803003	61	...
...	...	...

SUBJECT	ID	SCORE
解析 A	1803001	68
解析 A	1803003	61
...	...	...

Figure 3.5 Item-Based proper form of data

Python was used as the programing language to deal with the data because Python has many packages that contain coding and command that can be used easily for dealing with data such as pandas, NumPy, etc. Google Collaboratory was used as the platform for coding because it is easier to code from different places because it can access file stores online in sources like Google Drive and it can use a server from google to run and can increase its performance by setting. Its advantage was its runtime that cannot stand long computing time for the vast amount of data since our data was not that vast, it can be used for no problem.

3.3 Experiment for Analysis

In order to analyze the relation of diversity and accuracy, an experiment was designed based on the idea that “Less diversity of choices of the recommender system will lead to better accuracy of the recommended prediction” on the other hand “More diversity of choices of the recommender system will lead to less accuracy of the recommended prediction”. As it was described before, select subjects in each academic year were slightly different. There was some select subject that was unique only in that academic year and some select subject that was all able to be enrolled in all three

academic years. A number of select subjects that were available in all three academic years were less than a number of select subjects that were uniquely available in each academic year so the set of select subjects having in all 3 academic years is less diversity in choices than the set of unique select subjects in each academic year. The dataset was separated into two groups as the base data for the developed system in order to perform the analysis for the relation of diversity and accuracy.

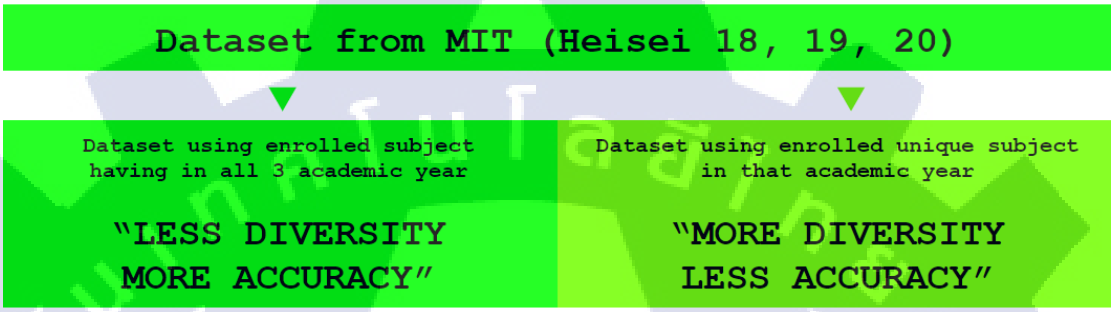


Figure 3.6 data set separation for analysis

Besides using two separate data sets for developing the recommender system to make the comparison analysis, the system will be developed by using several algorithms and similarity method computations as in Figure 3.7 in order to make the analysis.

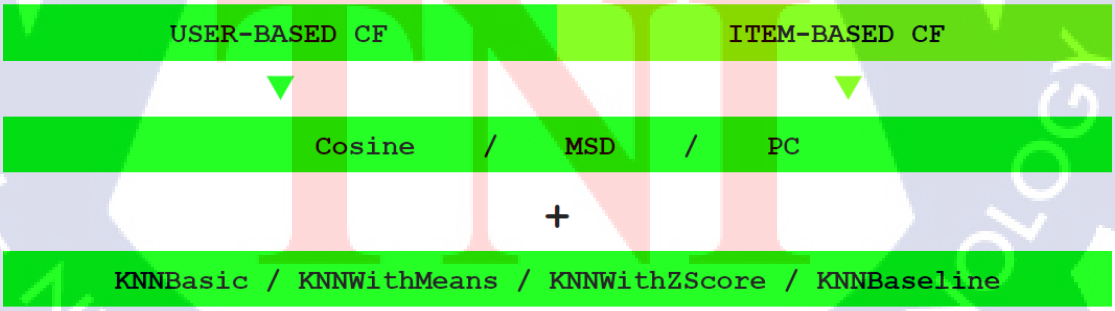


Figure 3.7 experiment design

By using KNN for collaborative filtering make system developing did not need to be concerned about the content of each subject. By using Surprise package. The similarities module includes tools to compute similarity metrics between users or items can be used as following;

3.3.1 surprise.similarities.cosine()

Compute the cosine similarity between all pairs of users (or items). Only common users (or items) are taken into account. The cosine similarity is defined as:

$$cosine\_sim(u,v) = \frac{\sum_{i \in \mathcal{J}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{J}_{uv}} r_{ui}^2 \sum_{j \in \mathcal{J}_{uv}} r_{ui}^2}} \quad \text{or}$$

$$cosine\_sim(i,j) = \frac{\sum_{u \in U_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U_{ij}} r_{ui}^2 \sum_{u \in U_{ij}} r_{uj}^2}}$$

3.3.2 surprise.similarities.msd()

Compute the Mean Squared Difference similarity between all pairs of users (or items). Only common users (or items) are taken into account. The Mean Squared Difference is defined as:

$$msd(u,v) = \frac{1}{|I_{uv}|} \cdot \sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2 \quad \text{or}$$

$$msd(i,j) = \frac{1}{|U_{ij}|} \cdot \sum_{u \in U_{ij}} (r_{ui} - r_{uj})^2$$

The MSD-similarity is then defined as:

$$msd\_sim(u,v) = \frac{1}{msd(u,v) + 1} \quad \text{or}$$

$$msd\_sim(i,j) = \frac{1}{msd(i,j) + 1}$$

3.3.3 surprise.similarities.pearson()

Compute the Pearson correlation coefficient between all pairs of users (or items). Only common users (or items) are taken into account. The Pearson correlation coefficient can be seen as a mean-centered cosine similarity, and is defined as:

$$pearson\_sim(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \mu_u)(r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \mu_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \mu_v)^2}} \quad \text{or}$$

$$pearson\_sim(i,j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \mu_i)(r_{uj} - \mu_j)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \mu_i)^2 \sum_{u \in U_{ij}} (r_{uj} - \mu_j)^2}}$$

Chapter 4
Result and discussion

In this section, the received result from experiments will be shown separating by academic year, method, and base of system sort by the performance got from the algorithm descending from the best to the worst prediction performance then compare with the dataset that use only subject that has in all 3 academic years to see the effect of diversity to accuracy of the recommender system. The results will be shown as following;

4.1 Results from experiment performed on developed recommender system

4.1.1 Results from developed system based on academic year of Heisei 18 data using enrolled unique subject of that academic year

4.1.1.1 User-based collaborative filtering system

4.1.1.2 Item-based collaborative filtering system

4.1.2 Results from developed system based on academic year of Heisei 18 data using enrolled subject having in all three academic year

4.1.2.1 User-based collaborative filtering system

4.1.2.2 Item-based collaborative filtering system

4.1.3 Results from developed system based on academic year of Heisei 19 data using enrolled unique subject of that academic year

4.1.3.1 User-based collaborative filtering system

4.1.3.2 Item-based collaborative filtering system

4.1.4 Results from developed system based on academic year of Heisei 19 data using enrolled subject having in all three academic year

4.1.4.1 User-based collaborative filtering system

4.1.4.2 Item-based collaborative filtering system

4.1.5 Results from developed system based on academic year of Heisei 20 data using enrolled unique subject of that academic year

4.1.5.1 User-based collaborative filtering system

4.1.5.2 Item-based collaborative filtering system

4.1 Results from experiment performed on developed recommender system

4.1.1 Results from developed system based on academic year of Heisei 18 data using enrolled unique subject of that academic year

4.1.1.1 User-based collaborative filtering system

Table 4.1 result from KNN User-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 18 data using enrolled unique subject of that academic year

Method	**RMSE**
KNNWithMeans	21.928973
KNNWithZScore	22.205391
KNNBaseline	22.453555
KNNBasic	24.915419

Table 4.2 result from KNN User-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 18 data using enrolled unique subject of that academic year

Method	**RMSE**
KNNWithMeans	21.896528
KNNBaseline	21.952130
KNNWithZScore	22.025571
KNNBasic	23.991819

Table 4.3 result from KNN User-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 18 data using enrolled unique subject of that academic year

Method	**RMSE**
KNNWithMeans	21.887683
KNNWithZScore	21.927511
KNNBaseline	22.189867
KNNBasic	24.310877

4.1.1.2 Item-based collaborative filtering system

Table 4.4 result from KNN Item-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 18 data using enrolled unique subject of that academic year

Method	RMSE
KNNBaseline	21.695943
KNNWithMeans	22.068894
KNNWithZScore	22.256075
KNNBasic	24.245985

Table 4.5 result from KNN Item-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 18 data using enrolled unique subject of that academic year

Method	RMSE
KNNWithMeans	23.276065
KNNBaseline	23.420426
KNNWithZScore	23.686892
KNNBasic	24.628059

Table 4.6 result from KNN Item-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 18 data using enrolled unique subject of that academic year

Method	RMSE
KNNBaseline	21.988594
KNNWithMeans	22.181433
KNNWithZScore	22.203304
KNNBasic	24.297532

4.1.2 Results from developed system based on academic year of Heisei 18 data using enrolled subject having in all three academic year

4.1.2.1 User-based collaborative filtering system

Table 4.7 result from KNN User-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 18 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithMeans	21.965144
KNNBaseline	22.362417
KNNWithZScore	22.409084
KNNBasic	24.685384

Table 4.8 result from KNN User-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 18 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithMeans	21.750160
KNNBaseline	22.050195
KNNWithZScore	22.138681
KNNBasic	24.249471

Table 4.9 result from KNN User-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 18 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithMeans	21.941792
KNNWithZScore	22.098803
KNNBaseline	22.381596
KNNBasic	24.759964

4.1.2.2 Item-based collaborative filtering system

Table 4.10 result from KNN Item-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 18 data using enrolled subject having in all three academic year

Method	RMSE
KNNBaseline	21.917385
KNNWithMeans	22.263208
KNNWithZScore	22.410853
KNNBasic	24.433991

Table 4.11 result from KNN Item-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 18 data using enrolled subject having in all three academic year

Method	RMSE
KNNBaseline	23.438961
KNNWithZScore	23.505702
KNNWithMeans	23.521328
KNNBasic	25.586224

Table 4.12 result from KNN Item-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 18 data using enrolled subject having in all three academic year

Method	RMSE
KNNBaseline	21.835898
KNNWithMeans	22.270806
KNNWithZScore	22.486739
KNNBasic	24.652942

4.1.3 Results from developed system based on academic year of Heisei 19 data using enrolled unique subject of that academic year

4.1.3.1 User-based collaborative filtering system

Table 4.13 result from KNN User-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 19 data using enrolled unique subject of that academic year

Method	RMSE
KNNWithMeans	23.004966
KNNBaseline	23.247251
KNNWithZScore	23.404830
KNNBasic	26.062098

Table 4.14 result from KNN User-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 19 data using enrolled unique subject of that academic year

Method	RMSE
KNNBaseline	22.679866
KNNWithMeans	22.854448
KNNWithZScore	23.657296
KNNBasic	24.712529

Table 4.15 result from KNN User-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 19 data using enrolled unique subject of that academic year

Method	RMSE
KNNWithMeans	23.002475
KNNWithZScore	23.289208
KNNBaseline	23.391511
KNNBasic	25.285284

4.1.3.2 Item-based collaborative filtering system

Table 4.16 result from KNN Item-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 19 data using enrolled unique subject of that academic year

Method	RMSE
KNNBaseline	23.065433
KNNWithZScore	23.255827
KNNWithMeans	23.294823
KNNBasic	25.492821

Table 4.17 result from KNN Item-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 19 data using enrolled unique subject of that academic year

Method	RMSE
KNNWithMeans	24.531318
KNNBaseline	24.630276
KNNWithZScore	24.635005
KNNBasic	26.363738

Table 4.18 result from KNN Item-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 19 data using enrolled unique subject of that academic year

Method	RMSE
KNNBaseline	22.862440
KNNWithZScore	23.178340
KNNWithMeans	23.450824
KNNBasic	25.343633

4.1.4 Results from developed system based on academic year of Heisei 19 data using enrolled subject having in all three academic year

4.1.4.1 User-based collaborative filtering system

Table 4.19 result from KNN User-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 19 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithMeans	22.835089
KNNWithZScore	23.031967
KNNBaseline	23.205847
KNNBasic	25.860798

Table 4.20 result from KNN User-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 19 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithMeans	22.860989
KNNBaseline	22.865726
KNNWithZScore	23.203863
KNNBasic	24.780752

Table 4.21 result from KNN User-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 19 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithMeans	22.827946
KNNWithZScore	23.111260
KNNBaseline	23.196114
KNNBasic	25.170503

4.1.4.2 Item-based collaborative filtering system

Table 4.22 result from KNN Item-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 19 data using enrolled subject having in all three academic year

Method	RMSE
KNNBaseline	22.838791
KNNWithMeans	23.057259
KNNWithZScore	23.171060
KNNBasic	25.347697

Table 4.23 result from KNN Item-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 19 data using enrolled subject having in all three academic year

Method	RMSE
KNNBaseline	24.100500
KNNWithZScore	24.349285
KNNWithMeans	24.637676
KNNBasic	26.271755

Table 4.24 result from KNN Item-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 19 data using enrolled subject having in all three academic year

Method	RMSE
KNNBaseline	22.809535
KNNWithMeans	22.915273
KNNWithZScore	23.341569
KNNBasic	25.246148

4.1.5 Results from developed system based on academic year of Heisei 20 data using enrolled unique subject of that academic year

4.1.5.1 User-based collaborative filtering system

Table 4.25 result from KNN User-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 20 data using enrolled unique subject of that academic year

Method	RMSE
KNNWithMeans	22.685775
KNNBaseline	22.765060
KNNWithZScore	22.797698
KNNBasic	25.567919

Table 4.26 result from KNN User-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 20 data using enrolled unique subject of that academic year

Method	RMSE
KNNBaseline	22.662944
KNNWithMeans	22.764847
KNNWithZScore	22.968794
KNNBasic	24.913656

Table 4.27 result from KNN User-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 20 data using enrolled unique subject of that academic year

Method	RMSE
KNNWithZScore	22.396398
KNNWithMeans	22.825337
KNNBaseline	22.860783
KNNBasic	25.620572

4.1.5.2 Item-based collaborative filtering system

Table 4.28 result from KNN Item-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 20 data using enrolled unique subject of that academic year

Method	RMSE
KNNBaseline	22.385202
KNNWithZScore	22.703372
KNNWithMeans	22.814718
KNNBasic	24.338323

Table 4.29 result from KNN Item-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 20 data using enrolled unique subject of that academic year

Method	RMSE
KNNWithZScore	23.721141
KNNWithMeans	24.086012
KNNBaseline	24.251955
KNNBasic	25.620042

Table 4.30 result from KNN Item-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 20 data using enrolled unique subject of that academic year

Method	RMSE
KNNBaseline	22.620788
KNNWithZScore	22.729615
KNNWithMeans	22.802734
KNNBasic	24.652893

4.1.6 Results from developed system based on academic year of Heisei 20 data using enrolled subject having in all three academic year

4.1.6.1 User-based collaborative filtering system

Table 4.31 result from KNN User-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 20 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithZScore	22.682131
KNNWithMeans	22.837497
KNNBaseline	23.065624
KNNBasic	25.697611

Table 4.32 result from KNN User-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 20 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithZScore	22.656641
KNNBaseline	22.702100
KNNWithMeans	22.712458
KNNBasic	24.724217

Table 4.33 result from KNN User-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 20 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithMeans	22.642317
KNNWithZScore	22.684192
KNNBaseline	22.862096
KNNBasic	25.062021

4.1.6.2 Item-based collaborative filtering system

Table 4.34 result from KNN Item-Based collaborative filtering using Cosine similarity computation method based on academic year of Heisei 20 data using enrolled subject having in all three academic year

Method	RMSE
KNNBaseline	22.413554
KNNWithMeans	22.745283
KNNWithZScore	22.875081
KNNBasic	24.401797

Table 4.35 result from KNN Item-Based collaborative filtering using MSD similarity computation method based on academic year of Heisei 20 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithMeans	23.518488
KNNWithZScore	23.728754
KNNBaseline	24.033841
KNNBasic	25.726542

Table 4.36 result from KNN Item-Based collaborative filtering using Pearson Correlation similarity computation method based on academic year of Heisei 20 data using enrolled subject having in all three academic year

Method	RMSE
KNNWithMeans	22.501073
KNNBaseline	22.545635
KNNWithZScore	22.784087
KNNBasic	24.640243

After we got the result, the analysis of diversity and accuracy in the recommender system will be conducted. Firstly, the analysis of comparing similarity computation method will be conducted in order to see which method will provide the best prediction accuracy and is there any effect of diversity to accuracy in each method. Secondly, the analysis of comparing algorithm of KNN method will be conducted in order to see which method will provide the best prediction accuracy and is there any effect of diversity to accuracy in each method.

4.2 Discussion

4.2.1 Analysis of similarity computation method

First, we will compare average RMSE of each method perform on both user-based and item-based to see what similarity method give us the best performance

Table 4.37 the comparison analysis of RMSE in case of user-based

Method	Cosine Similarity	MSD Similarity	Pearson Correlation
Heisei 18	22.8758345	22.466512	22.5789845
Heisei 19	23.92978625	23.47603475	23.7421195
Heisei 20	23.454113	23.32756025	23.4257725

Table 4.37 the comparison analysis of RMSE in case of user-based (Continued)

Method	Cosine Similarity	MSD Similarity	Pearson Correlation
Heisei 18 using all 3	22.85550725	22.54712675	22.79553875
Heisei 19 using all 3	23.73342525	23.4278325	23.57645575
Heisei 20 using all 3	23.57071575	23.198854	23.3126565

We found that MSD similarity is the best similarity method calculation that will give the best performance for prediction so we will use this method for the analysis and ignore other method as for the developing of the recommender system, the best similarity method will only be considered.

Table 4.38 the comparison analysis of RMSE User-based using MSD similarity

Method Year Data	Average RMSE	Using all 3 subjects Average RMSE
Heisei 18	22.466512	22.54712675
Heisei 19	23.47603475	23.4278325
Heisei 20	23.32756025	23.198854

In case of using MSD Similarity in user-based recommender system, we found that 2 in 3 of recommender system ; Heisei 19 and Heisei 20 that use only recommended subject that have in all 3 academic years is perform better than the one that use all recommended subject in its year which has more numbers of subjects which mean more diversity. Only Heisei 18 that is not follow the hypothesis, more diverse, less accuracy.

So we will focus on Heisei 18 to see what caused it to not follow the hypothesis. The subject that only in Heisei 18 information are the causes that we firstly think.

Table 4.39 the information of select enrolled unique subject of that academic year Heisei 18

Coures' Name	Number of attended undergraduates
データの統計解析	90
データ統計解析応用演習	89
システム制御工学	1
ゼミナール「環境と社会」	1
国際関係論	4
ゼミナール「市民と公共」Ａ	2
ゼミナール「市民と公共」Ｂ	1
ゼミナール「思考と数理」Ｂ	1
日本語 A-1	1
日本語 A-2	1
日本語 B-2	1
日本語 C-2	1

Due to the number of available students that attend in subject "データの統計解析" and "日本の歴史" are high, can be calculated as 93% and 92%. It can cause the

prediction performance to be better due to having much more data as references for similarity calculation.

Table 4.40 the comparison analysis of RMSE in case of item-based

Method	Cosine Similarity	MSD Similarity	Pearson Correlation
Heisei 18	22.56672425	23.7528605	22.66771575
Heisei 19	23.777226	25.04008425	23.70880925
Heisei 20	23.06040375	24.4197875	23.2015075
Heisei 18 using all 3	22.75635925	24.01305375	24.01305375
Heisei 19 using all 3	23.60370175	24.839804	24.839804
Heisei 20 using all 3	23.10892875	24.25190625	24.25190625

We found that cosine similarity is the best similarity method calculation that will give the best performance for prediction so we will use this method for the analysis and ignore other method as for the developing of the recommender system, the best similarity method will only be considered.

Table 4.41 the comparison analysis of RMSE item-based using cosine similarity

Method Year Data	Average RMSE	Using all 3 subjects Average RMSE
Heisei 18	22.56672425	22.75635925
Heisei 19	23.777226	23.60370175
Heisei 20	23.06040375	23.10892875

In case of using cosine similarity in item-based recommender system, we found that only 1 in 3 of recommender system ; Heisei 19 that use only recommended subject that have in all 3 academic years is perform better than the one that use all recommended subject in its year which has more numbers of subjects which mean more diversity. Both Heisei 18 and Heisei 20 are not follow the hypothesis, more diverse, less accuracy.

As we know that item-based use the similarity between subjects to make a recommendation so in Heisei 18 and Heisei 20 that have more subjects to use as the reference, the system performance in prediction can be done better. We will focus in Heisei 19 that is the only one that follows the hypothesis.

Table 4.42 the information of select enrolled unique subject of that academic year Heisei 19

Coures' Name	**Number of attended undergraduates**
情報計測工学	85
データの統計解析	98
データ統計解析応用演習	107
海外語学研修	1
地域再生システム論	8
ゼミナール「環境と社会」	1
日本近現代史Ｂ	8
ゼミナール「市民と公共」Ａ	3
日本語 A-1	2
日本語 A-2	2

Table 4.42 the information of select enrolled unique subject of that academic year Heisei 19 (Continued)

Coures' Name	Number of attended undergraduates
日本語 B-2	2
日本語 D-1	1

Even the number of available students that attend in subjects "情報計測工学", "情報計測工学", and "データ統計解析応用演習" are high, can be calculated as 77%, 88%, and 96%, in this case of user-based system, it means almost all students enroll in these subjects so the similarity between these subjects don't help improve the efficient in making the prediction process of the system that much.

4.2.2 Analysis of algorithm method

First, we will compare average RMSE of each method perform on both user-based and item-based to see what similarity method give us the best performance

Table 4.43 the comparison analysis of RMSE in case of user-based

Method	KNNBasic	KNNWith Means	KNNWith ZScore	KNNBaseline
Heisei 18	24.40603833	21.90439467	22.05282433	22.19851733
Heisei 19	25.35330367	22.953963	23.45044467	23.10620933
Heisei 20	25.36738233	22.758653	22.72096333	22.762929
Heisei 18 using all 3	24.56493967	21.88569867	22.21552267	22.264736

Table 4.43 the comparison analysis of RMSE in case of user-based (Continued)

Method	KNNBasic	KNNWith Means	KNNWith ZScore	KNNBaseline
Heisei 19 using all 3	25.27068433	22.84134133	23.11569667	23.089229
Heisei 20 using all 3	25.161283	22.73075733	22.67432133	22.87660667

We found that KNN with Means is the algorithm method that will give the best performance for prediction for most of the systems so we will use this method for the analysis and ignore other method as for the developing of the recommender system, the best similarity method will only be considered.

Table 4.44 the comparison analysis of RMSE User-based using KNNWithMeans

Method Year Data	Average RMSE	Using all 3 subjects Average RMSE
Heisei 18	21.90439467	21.88569867
Heisei 19	22.953963	22.84134133
Heisei 20	22.758653	22.73075733

In case of using KNNWithMeans in user-based recommender system, we found that all recommender system ; Heisei 18, Heisei 19, and Heisei 20 that use only recommended subject that have in all 3 academic years is perform better than the one that use all recommended subject in its year which has more numbers of subjects which mean more diversity.

Table 4.45 the comparison analysis of RMSE in case of item-based

Method	KNNBasic	KNNWithMeans	KNNWithZScore	KNNBaseline
Heisei 18	24.39052533	22.50879733	22.71542367	22.368321
Heisei 19	25.73339733	23.75898833	23.689724	23.519383
Heisei 20	24.87041933	23.234488	23.051376	23.08598167
Heisei 18 using all 3	24.89105233	22.685114	22.801098	22.39741467
Heisei 19 using all 3	25.62186667	23.536736	23.620638	23.24960867
Heisei 20 using all 3	24.92286067	22.92161467	23.12930733	22.99767667

We found that KNN Baseline is the algorithm method that will give the best performance for prediction for most of the systems so we will use this method for the analysis and ignore other method as for the developing of the recommender system, the best similarity method will only be considered.

Table 4.46 the comparison analysis of RMSE Item-based using KNNBaseline

Method Year Data	Average RMSE	Using all 3 subjects Average RMSE
Heisei 18	22.368321	22.39741467
Heisei 19	23.519383	23.24960867
Heisei 20	23.08598167	22.99767667

In case of using KNN Baseline in item-based recommender system, we found that 2 in 3 of recommender system ; Heisei 19 and Heisei 20 that use only recommended subject that have in all 3 academic years is perform better than the one that use all recommended subject in its year which has more numbers of subjects which mean more diversity. Only Heisei 18 that is not follow the hypothesis, more diverse, less accuracy.

So we will focus on Heisei 18 to see what caused it to not follow the hypothesis. Because KNN Baseline take baseline rating (β_{ui}) into computing the rating so the baseline rating is the cause for the act of not following the hypothesis. Yuheda [78] stated that typical CF data exhibit large user and item effects systematic tendencies for some users to give higher ratings than others—and for some items to receive higher ratings than others. It is customary to adjust the data by accounting for these effects, which he encapsulated within the baseline estimates (β_{ui}). Denote by μ the overall average rating. A baseline estimate for an unknown rating r_{ui} is denoted by b_{ui} and accounts for the user and item effects:

$$b_{ui} = \mu + b_u + b_i$$

The parameters b_u and bi indicate the observed deviations of user u and item i, respectively, from the average. For example, suppose that we want a baseline estimate for the rating of the movie Titanic by user Joe. Now, say that the average rating over all movies, μ, is 3.7 stars. Furthermore, Titanic is better than an average movie, so it tends to be rated 0.5 stars above the average. On the other hand, Joe is a critical user, who tends to rate 0.3 stars lower than the average. Thus, the baseline estimate for Titanic's rating by Joe would be 3.9 stars by calculating 3.7 − 0.3 + 0.5. In this case, we focus on the subjects that only have in Heisei 18.

Table 4.47 the information of select enrolled unique subject of that academic year Heisei 18

Subject Name	Number of attended undergraduates
システム制御工学	1
ゼミナール「環境と社会」	1
国際関係論	4
ゼミナール「市民と公共」A	2
ゼミナール「市民と公共」B	1
ゼミナール「思考と数理」B	1
日本語 A-1	1
日本語 A-2	1
日本語 B-2	1
日本語 C-2	1

Due to the number of available students that attend in subject “データの統計解析” and “日本の歴史” are high, can be calculated as 93% and 92%. It can cause the prediction performance to be better due to having much more data as item deviation for baseline rating calculation.

An analysis showed that using MSD similarity as the similarity calculation method and KNNWithMeans as the algorithm will give the best prediction result for the user-based system. The hypothesis that stated more diversity will cause the accuracy to fall can still be applied but in some cases that the number of data that was reduced in order to make the diversity less is high. It can ignore the effect of diversity and make the prediction performance of the system that use to be better due to having much more data as references for calculation.

Chapter 5
Conclusion and Future works

Conclusion and future works of this study will be discussed as following;

5.1 Conclusion

5.2 Future works

5.1 Conclusion

An analysis showed that using MSD similarity as the similarity calculation method and KNNWithMeans as the algorithm will give the best prediction result for the user-based system. The hypothesis that stated more diversity will cause the accuracy to fall can still be applied but in some cases that the number of data that was reduced in order to make the diversity less is high. It can ignore the effect of diversity and make the prediction performance of the system that use to be better due to having much more data as references for calculation.

For the item-based system, using cosine similarity as the similarity calculation method and KNN Baseline as the algorithm will give the best prediction result. The hypothesis that stated more diversity will cause the accuracy to fall cannot be applied because as we know that item-based use the similarity between subjects to make a recommendation so the system that have more subjects to use as the reference, the system performance in prediction can be done better. In some cases, having more choices for a recommended item may not lead to better prediction result due to similar recommended item cannot be used much in the item similarity calculation.

This research study was conducted in order to discover the base knowledge that can be set as a standard or a case study for developing the recommendation system in the field of education because there was not much pieces of research in this topic in terms of both the recommender system in the field of education and the diversity effect for accuracy based on the real dataset that full of biases and outliers. Hoping this research can be used as the start line for developing the recommender system for the education field.

5.2 Future works

For future works following this study, Muroran Institutes of Technology's undergraduates can register enrollable subject on web page system called "Moodle". This step of enrolling and registering can be shown in Figure 5.1 through 5.4 Start from select academic year then select your wanted subject. The subject that cannot be enrolled will not be shown. The recommender system can be deployed in the final screen for selection to enroll it as the additional text to warn or recommend that undergrad that he/she should or should not enroll in that subject.

Figure 5.1 Moodle System home screen

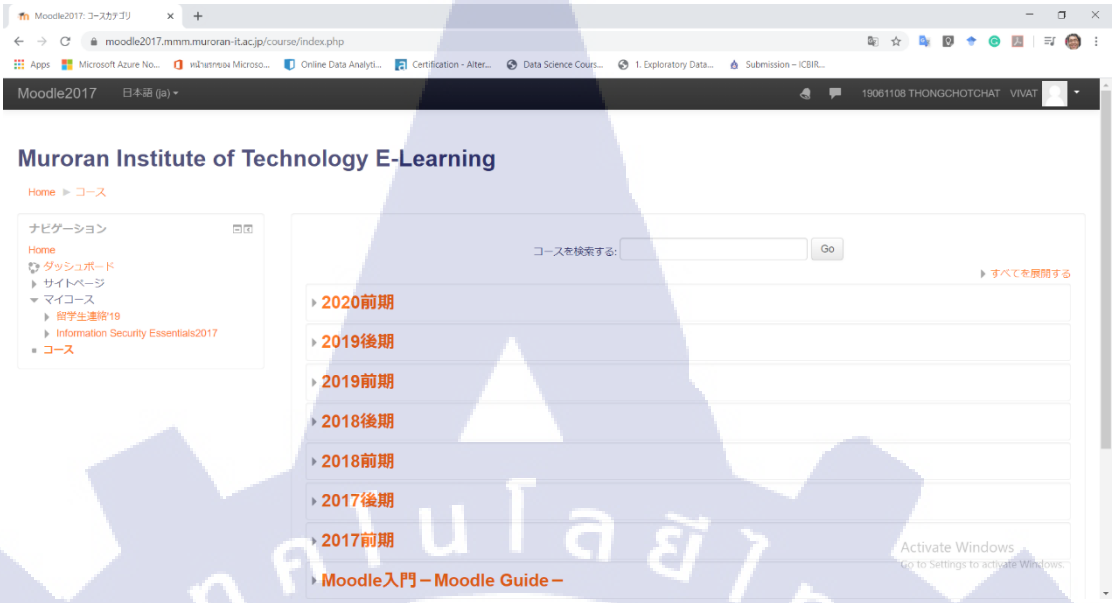


Figure 5.2 Moodle System academic year selection

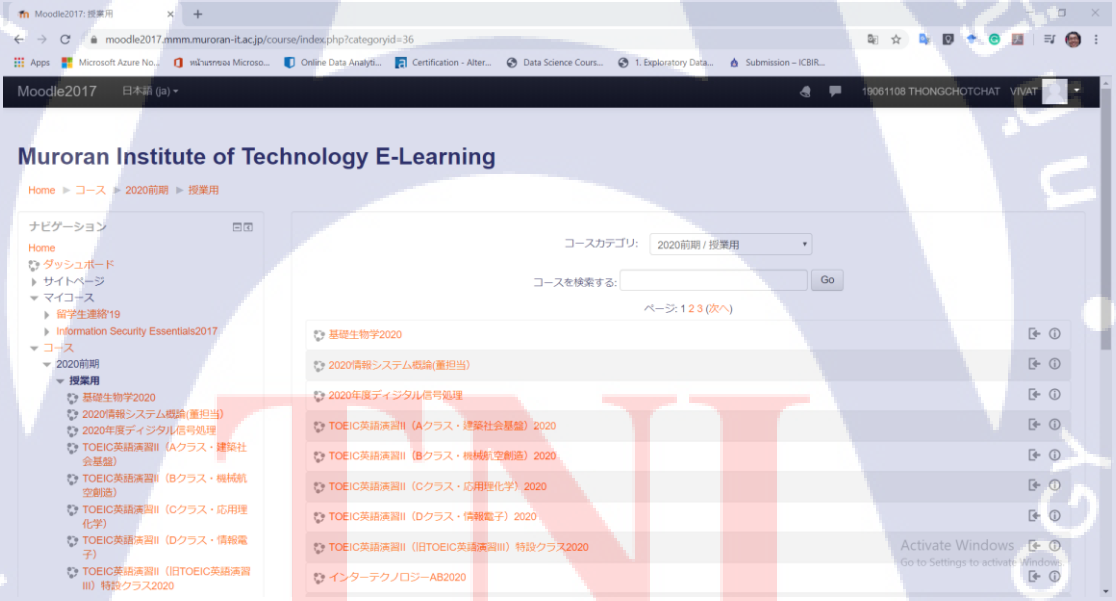


Figure 5.3 Moodle System subject selection

Figure 5.4 the recommender system can be deployed in this scenario as suggestion message

Reference

Reference

[1] Michael D. Ekstrand et al., "Collaborative filtering recommender systems," *Foundations and Trends in Human-Computer Interaction,* vol. 4, no. 2, pp. 81-173, Febuary 2011.

[2] J. Bennett and S. Lanning, "The netflix prize," *Proceedings of KDD Cup and Workshop 2007,* San Jose, California, USA, August 12, 2007, pp. 35.

[3] Amazon.com, "Q4 2009 financial results," (*Earnings Report Q4-2009)*, USA: Amazon.com, Inc., 2010.

[4] B. Schwartz, *The Paradox of Choice: Why More is Less,* New York: ECCO, 2004.

[5] M. Balabanovic and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM,* vol. 40, no. 3, pp. 66–72, March 1997.

[6] D. Goldberg et al., "Using collaborative filtering to weave an information tapestry," *Communications of the ACM,* vol. 35, no. 12, pp. 61–70, December 1992.

[7] P. Resnick et al., "GroupLens: an open architecture for collaborative filtering of netnews," *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW94,* ACM, New York, USA, October 22, 1994, pp. 175–186.

[8] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating 'word of mouth,' " *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'95,* Denver, Colorado, USA, May 1, 1995, pp. 210–217.

[9] C. Vialardi-Sacín et al., "Recommendation in higher education using data mining techniques," *Proceedings of 2nd International Working Group on Educational Data Mining 2009, EDM'09,* Cordoba, Spain, July 1-3, 2009, pp. 190-199.

[10] J.R. Quinlan, *C4.5: Programs for Machine Learning*, California: Morgan Kaufmann Publishers, 1993.

[11] J.F. Superby et al, "Determination of factors influencing the achievement of the first-year university students using data mining methods," *Workshop on Educational Data Mining*, vol. 32, p. 37-44, June 2006.

[12] K.I. Ghauth and N.A. Abdullah, "Learning materials recommendation using good learners' ratings and content-based filtering," *Educational Technology Research and Development*, vol. 58, no. 6, pp. 711-727, December 2010.

[13] C.N. Ziegler et al., "Improving recommendation lists through topic diversification," *Proceedings of the 14th International Conference on World Wide Web, WWW 2005*, ACM, New York, NY, USA, May 10, 2005, pp. 22–32.

[14] P. Adamopoulos and A. Tuzhilin, "On unexpectedness in recommender systems: or how to better expect the unexpected," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 4, pp. 1–32, December 2014.

[15] G. Adomavicius and Y. Kwon, "Improving aggregate recommendation diversity using ranking- based techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 896–911, May 2012.

[16] O. Celma and P. Herrera, "A new approach to evaluating novel recommendations," *Proceedings of the 2nd ACM Conference on Recommender Systems, RecSys 2008*, ACM, New York, NY, USA, October 23, 2008, pp. 179–186.

[17] N. Hurley and M. Zhang, "Novelty and diversity in top-N recommendation – analysis and evaluation," *ACM Transactions on Internet Technology*, vol. 10, no. 4, pp. 1–30, March 2011.

[18] S. Vargas et al., "Rank and relevance in novelty and diversity metrics for recommender systems," *Proceedings of the 5th ACM Conference on Recommender Systems, RecSys 2011*, ACM, New York, NY, USA, October 21, 2011, pp. 109–116.

[19] L. McAlister and E.A. Pessemier, "Variety seeking behavior: an interdisciplinary review," *Journal of Consumer Research*, vol. 9, no. 3, pp. 311–322, December 1982.

[20] S.R. Maddi, *The Pursuit of Consistency and Variety*, Chicago: Rand McNally, 1968.

[21] P.S. Raju, "Optimum stimulation level: its relationship to personality, demographics and exploratory behavior," *Journal of Consumer Research*, vol. 7, no. 3, pp. 272–282, December 1980.

[22] Python Software Foundation, *"What is Python? Executive Summary,"* [Online]. Available: https://www.python.org/doc/essays/blurb/. [Accessed: January 25, 2020]

[23] Python Software Foundation, *"6. Modules,"* The Python Tutorial [Online]. Available: https://docs.python.org/3/tutorial/modules.html. [Accessed: January 25, 2020]

[24] Google, *"Frequently Asked Questions,"* [Online]. Available: https://research.google.com/colaboratory/faq.html. [Accessed: January 25, 2020]

[25] R. Burke, *The Adaptive Web*, Berlin / Heidelberg: Springer, 2007.

[26] P. Resnick and H.R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, March 1997.

[27] G. Linden et al., "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, January 2003.

[28] M. Montaner et al., "A taxonomy of recommender agents on the internet," *Artificial Intelligence Review*, vol. 19, no. 4, pp. 285–330, June 2003.

[29] D. Billsus and M. Pazzani, *"Learning Probabilistic User Models,"* [Online]. Available: http://www.dfki.de/~bauer/um-ws/. [Accessed: December 15, 2019]

[30] G. Fisher, "User modeling in human-computer interaction," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1-2, pp. 65–86, March 2001.

[31] S. Berkovsky et al., "Mediation of user models for enhanced personalization in recommender systems," *User Modeling and User-Adapted Interaction*, vol. 18, no. 3, pp. 245–286, August 2008.

[32] S. Berkovsky et al., "Cross-representation mediation of user models," *User Modeling and User-Adapted Interaction*, vol. 19, no. 1–2, pp. 35–63, February 2009.

[33] N. Taghipour et al., "Usage-based web recommendations: a reinforcement learning approach," *Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys 2007*, Minneapolis, MN, USA, October 19–20, 2007, pp. 113–120.

[34] T. Mahmood et al., "Improving recommendation effectiveness by adapting the dialogue strategy in online travel planning," *International Journal of Information Technology and Tourism*, vol. 11, no. 4, pp. 285–302, December 2009.

[35] R. Burke, *The Adaptive Web*, Berlin / Heidelberg: Springer, 2007.

[36] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, April 2005.

[37] F. Ricci, *Recommender Systems: Models and Techniques,* New York: Springer, 2014.

[38] M. Balabanovic and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, March 1997.

[39] D. Billsus and M.J. Pazzani, "User modeling for adaptive news access," *User Modeling and User-Adapted Interaction*, vol. 10, no. 2–3, pp. 147 - 180, June 2000.

[40] K. Lang, "News weeder: learning to filter netnews." *Proceedings of the 12th International Conference on Machine Learning, ICML,* Tahoe City, California, July 9–12, 1995, pp. 331 - 339.

[41] M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Machine Learning*, vol. 27, no. 3, pp. 313 - 331, June 1997.

[42] D. Michael et al., "Collaborative filtering recommender systems," *Foundations and Trends in Human-Computer Interaction*, vol. 4, no. 2, pp. 81-173, February 2011

[43] G. Adomavicius and A. Tuzhilin, *Context-Aware Recommender Systems,* (Recommender Systems Handbook), Boston, MA: Springer, 2011.

[44] J. Delgado and N. Ishii, "Memory-based weighted majority prediction for recommender systems," *Proceedings of the ACM SIGIR'99 Workshop on Recommender Systems, SIGIR'99*, Berkeley, California, USA, August 15, 1999.

[45] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *ACMTransaction on Information Systems*, vol. 22, no. 1, pp. 143–177, January 2004.

[46] W. Hill et al., “Recommending and evaluating choices in a virtual community of use,” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’95,* New York, NY, USA, May 1, 1995, pp. 194–201.

[47] J.A. Konstan et al., “GroupLens: applying collaborative filtering to usenet news,” *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, Mar 1997.

[48] A. Nakamura, “Collaborative filtering using weighted majority prediction algorithms,” *Proceedings of the 15th International Conference on Machine Learning, ICML ’98*, Madison, Wisconsin, USA, July 24-27, 1998, pp. 395–403.

[49] B. Sarwar et al., “Item-based collaborative filtering recommendation algorithms,” *Proceedings of the 10th International Conference on World Wide Web, WWW ’01,* ACM, New York, NY, USA, April 1, 2001, pp. 285–295.

[50] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’08,* ACM, New York, NY, USA, August 24, 2008, pp. 426–434.

[51] G. Takács et al., “Major components of the gravity recommendation system,” *SIGKDD Exploration Newsletter*, vol. 9, no. 2, pp. 80–83, December 2007.

[52] J.S. Breese et al., “Empirical analysis of predictive algorithms for collaborative filtering,” *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence, UAI1980,* Morgan Kaufmann, San Mateo, California, USA, January 30, 1998, pp. 43–52.

[53] J.L. Herlocker et al., “An algorithmic framework for performing collaborative filtering,” *Proceedings of the 22nd Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’99,* ACM, New York, NY, USA, August 2, 2017, pp. 227–234.

[54] A.E. Howe and R.D. Forbes, “Re-considering neighborhood-based collaborative filtering parameters in the context of new data,” *Proceeding of the 17th ACM Conference on Information and Knowledge Management, CIKM ’08,* ACM, New York, NY, USA, October 26, 2008, pp. 1481–1482.

[55] W.W. Cohen et al., “Learning to order things,” *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems, NIPS ’97,* Cambridge, MA, USA, December 1-6, 1997, pp. 451–457.

[56] Y. Freund et al., “An efficient boosting algorithm for combining preferences,” *Journal of Machine Learning Research*, vol. 4, no. 6, pp. 933–969, January 2003.

[57] R. Jin et al., “Preference-based graphic models for collaborative filtering,” *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence, UAI’03*, Morgan Kaufmann, San Francisco, CA, October 19, 2003, pp. 329–33.

[58] R. Jin et al., “Collaborative filtering with decoupled models for preferences and ratings,” *Proceedings of the 12th International Conference on Information and Knowledge Management, CIKM ’03*, ACM, New York, NY, USA, November 3, 2003, pp. 309–316.

[59] J. Herlocker et al., “An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms,” *Information Retrieval*, vol. 5, no. 4, pp. 287–310, October 2002.

[60] S.M. McNee et al., “Being accurate is not enough: how accuracy metrics have hurt recommender systems,” *Proceedings of CHI 2006 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2006*, ACM, New York, NY, USA, April 21, 2006, pp. 1097–1101.

[61] C.H. Coombs and G.S. Avrunin, “Single peaked preference functions and theory of preference,” *Psychological Review*, vol. 84, no. 2, pp. 216–230, March 1977.

[62] P. Brickman and B. D’Amato, “Exposure effects in a free-choice situation,” *Journal of Personality and Social Psychology*, vol. 32, no. 3, pp. 415–420, September 1975.

[63] E. Pariser, *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think*, London: Penguin Books, 2012.

[64] M. Lubatkin and S. Chatterjee, “Extending modern portfolio theory into the domain of corporate diversification: does it apply?,” *The Academy of Management Journal*, vol. 37, no. 1, pp. 109–136, February 1994.

[65] C. Anderson, *The Long Tail: Why the Future of Business Is Selling Less of More*, Westport, Connecticut: Hyperion, 2006.

[66] G. Adomavicius and Y. Kwon, “Optimization-based approaches for maximizing aggregate recommendation diversity,” *INFORMS Journal on Computing*, vol. 26, no. 2, pp. 351–369, May 2014.

[67] T. Zhou et al., "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences, PNAS,* USA, March 9, 2010, pp. 4511 – 4515.

[68] A. Bellogín et al., "A comparative study of heterogeneous item recommendations in social systems," *Information Sciences,* vol. 221, pp. 142–169, February 2013.

[69] N. Lathia et al., "Temporal diversity in recommender systems," *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010,* ACM, New York, NY, USA, July 19, 2010, pp. 210–217.

[70] G.P. Patil and C. Taillie, "Diversity as a concept and its measurement," *Journal of the American Statistical Association,* vol. 77, no. 379, pp. 548–561, September 1982.

[71] T. Murakami at el., "Metrics for evaluating the serendipity of recommendation lists," *New Frontiers in Artificial Intelligence, Lecture Notes in Computer Science,* vol. 4914, pp. 40 - 46, June 2007.

[72] Y.C. Zhang et al., "Auralist: Introducing serendipity into music recommendation," *Proceedings of the 5th ACM Conference on Web Search and Data Mining, WSDM 2012,* ACM, New York, NY, USA, February 8, 2012, pp. 13–22.

[73] D.M. Fleder and K. Hosanagar, "Blockbuster culture's next rise or fall: the impact of recommender systems on sales diversity," *Management Science,* vol. 55, no. 5, pp. 697–712, March 2009.

[74] Z. Szlávik et al., "Diversity measurement of recommender systems under different user choice models," *Proceedings of the 5th AAAI Conference on Weblogs and Social Media, ICWSM 2011,* Barcelona, Spain, July 5, 2011, pp. 369-376.

[75] G. Adomavicius and Y. Kwon, "Overcoming accuracy-diversity tradeoff in recommender systems: a variance-based approach," *Proceedings of the 2008 Workshop on Information Technologies and Systems, WITS 2008,* Paris, France, January 1, 2008, pp. 151-156.

[76] A. Said et al., "Increasing diversity through furthest neighbor-based recommendation," *Proceedings of the WSDM Workshop on Diversity in Document Retrieval, DDR 2012,* Seattle, WA, USA, February 8-12, 2012, pp. 12-15.

[77] A. Said et al., “User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm,” *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, ACM, San Antonio Texas, USA, February 23, 2013, pp. 1399 – 1408.

[78] Y. Koren, “Factor in the neighbors: scalable and accurate collaborative filtering,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, pp. 1 - 24, January 2010.