

Глазунова Олена Григорівна*Національний університет біоресурсів та природокористування України, м. Київ*
ORCID 0000-0002-0136-4936**Понзель Ярослав Юрійович***Національний університет біоресурсів і природокористування України, м. Київ*
ORCID 0009-0005-6282-1888

МАТЕМАТИЧНА МОДЕЛЬ ОБРОБКИ ДАНИХ З ВИКОРИСТАННЯМ КОМБІНОВАНИХ МЕТОДІВ СПІЛЬНОЇ ФІЛЬТРАЦІЇ ТА МАТРИЧНОЇ ФАКТОРИЗАЦІЇ ДЛЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ В ОСВІТІ

Анотація. Ефективність рекомендаційних систем, які використовуються у різних галузях і, зокрема у освітній системі, є актуальною проблематикою для досліджень науковців та розробників програмних систем. На прикладі системи, яка надає рекомендації для вибору освітніх компонентів навчального плану на основі різних параметрів, зокрема попередніх навчальних досягнень студентів, даних щодо їх попереднього вибору дисциплін та вибору інших студентів тощо. Основною метою дослідження є розробка математичної моделі, яка здатна ефективно передбачати оцінки для предметів, які студенту даються на вибір.

У рамках поточної реалізації використовуються інструменти ML.NET, що автоматизують процес факторизації матриць та мінімізації функції втрат, спрощуючи розробку і налаштування моделі. Основні етапи реалізації включають тренування моделі через використання MatrixFactorizationTrainer, визначення параметрів сутності-тренера, а також оптимізацію моделі через градієнтний спуск та регуляризацию за допомогою норми Фробеніуса. Також застосовується функція втрат у вигляді середньоквадратичної помилки, що дозволяє оцінити точність передбачення та досягти мінімізації різниці між реальними та прогнозованими оцінками.

Запропоновано шляхи для покращення моделі, такі як: інтеграція контентної фільтрації, що дозволить нам в майбутньому враховувати додаткові характеристики предметів, такі як тип або ж складність, що значно підвищить точність рекомендацій. Також, існує можливість покращити обчислення схожості між предметами або студентами, що в свою чергу дозволить робити рекомендації для кінцевого користувача на основі подібності.

У підсумку, демонструється підхід, який дозволяє створити ефективну рекомендаційну систему, що може бути застосована для прогнозування результатів навчання студентів та формування персоналізованих освітніх програм. Підвищення точності таких систем відкриває нові можливості для покращення освітнього процесу, сприяючи більш індивідуальному підходу до навчання.

Ключові слова: рекомендаційна система, матрична факторизація, прогнозування оцінок, середньоквадратична помилка, корінь середньоквадратичної помилки, норма Фробеніуса, ML.NET, C#, контентна фільтрація, освіта, студент, індивідуальний план навчання.

Hlazunova Olena*National University of Life and Environmental Sciences of Ukraine, Kyiv*
ORCID 0000-0002-0136-4936**Ponzel Yaroslav***National University of Life and Environmental Sciences of Ukraine, Kyiv*
ORCID 0009-0005-6282-1888

MATHEMATICAL MODEL OF DATA COLLECTION AND PROCESSING FOR A RECOMMENDATION SYSTEM FOR FORMING AN INDIVIDUALIZED EDUCATIONAL PROGRAM

Abstract. *The effectiveness of recommendation systems, which are used in various fields and, in particular, in the educational system, is a relevant issue for research by scientists and software system developers. Using the example of a system that provides recommendations for selecting educational components of a curriculum based on various parameters, including previous academic achievements of students, data on their past course selections, and the choices of other students, etc. The main goal of the research is to develop a mathematical model capable of effectively predicting grades for courses that are offered to the student as optional.*

In the current implementation, ML.NET tools are used, which automate the process of matrix factorization and loss function minimization, simplifying the development and tuning of the model. The main stages of implementation include training the model using the MatrixFactorizationTrainer, defining the trainer's parameters, and optimizing the model through gradient descent and regularization by Frobenius Norm.

There are also some directions for improvement, such as integrating content-based filtering, which will allow us to take additional course characteristics, such as type or complexity, into account in the future, significantly improving the accuracy of recommendations. Alternatively, the calculation of similarity between courses or students can be enhanced, which would allow for making recommendations to end users based on the similarity.

In conclusion, an approach is demonstrated that allows for creating an effective recommendation system that can be used to predict student performance and form personalized educational programs. Improving the accuracy of such systems opens up new opportunities for enhancing the educational process, fostering a more individualized approach to learning.

Keywords: *recommendation system, matrix factorization, grade prediction, mean square error, root mean square error, Frobenius Norm, ML.NET, C#, content-based filtering, course selection, content-based filtering, education, student, individual study plan.*

1. Постановка проблеми

У наш час освітні системи стикаються з численними викликами, серед яких підтримка високої ефективності навчання, адаптація до індивідуальних потреб студентів та необхідність в персоналізації навчального процесу. Одним із основних моментів персонального навчання є проблема вибору навчальних предметів для студента. У багатьох випадках існуючі освітні платформи не мають інструментів для рекомендацій освітніх дисциплін, які могли б забезпечити максимальний результат для кожного студента, враховуючи його попередні досягнення, інтереси або ж схильності.

У зв'язку з цим, важливим кроком до підвищення якості навчання є побудова рекомендаційних систем для освітніх платформ. Один із способів вирішення такого роду проблеми є використання алгоритмів машинного навчання, щоб зможти надати персоналізовані рекомендації для студентів на основі визначених факторів. Однак, наявні методи потребують постійного вдосконалення, а також розробки математичних моделей, які б забезпечували точність разом із надійністю рекомендацій.

Дана стаття присвячена розробці математичної моделі збору і обробки даних для рекомендаційної системи вибору дисциплін для студента, що в свою чергу дозволить значно покращити процес навчання та підвищити ефективність освітніх платформ.

2. Аналіз останніх досліджень і публікацій

У статті [1] автори підмічають зв'язок між розвитком рекомендаційних систем та зростанням бізнесу в таких сферах, як Netflix, Amazon та Yahoo та підкреслюють важливість таких технологій як контентна фільтрація, колаборативна фільтрація, гібридні моделі, текстовий аналіз, кластеризація, матрична факторизація, нейронні мережі. Досліджено, що використання гібридних систем зростає, оскільки вони комбінують сильні сторони інших моделей.

У своїй роботі [2] розглядає приклад розробки системи для інтернет-магазину та демонструє математичну модель, яка розроблена на основі модифікаційного методу

колаборативної фільтрації і розв'язує багатокритеріальну задачу визначення рекомендацій для сервісу продажу. Подібність вподобань визначається за коефіцієнтом кореляції Пірсона.

У своїх напрацюваннях [3] автор розглядає точність передбачення для декількох типів рекомендаційних систем (інтелектуальної рекомендаційної системи, рекомендаційної системи на основі контенту, рекомендаційної системи за допомогою правил асоціації та рекомендаційної системи на основі колаборативного фільтрування). У висновку автор пропонує використовувати інтелектуальну рекомендаційну систему із використанням алгоритмів асоціацій. В роботі демонструються ефективність такої системи, швидкість виконання операцій нею та точність її передбачень.

В роботі [4] автори розглядають різні техніки та реалізації, що стосуються модельно-орієнтованих рекомендаційних систем. Досліджуються основні проблеми, пов'язані з рекомендаціями, включно з проблемою top-N рекомендацій, і досліджуються найсучасніші методи модельно-орієнтованих рекомендаційних систем, які використовуються для вирішення цих проблем. Авторами розглянуто методи систем рекомендацій від класичної матричної факторизації до найсучасніших підходів на основі GAN та зроблено висновок про те, що налаштовуючи різні параметри та розуміючи нюанси кожного методу, ми можемо використати їх потужність для надання індивідуальних високоякісних рекомендацій. Різноманітний набір моделей, які тут обгрунтовано, підкреслює необхідність підходу до нюансів у виборі правильної техніки рекомендацій, враховуючи такі фактори, як характеристики даних, вимоги до програми та обчислювальні ресурси.

В статті [5] пропонується структура рекомендаційної системи на основі спільної фільтрації, що інтегрує соціальні зв'язки між користувачами та їх взаємодії з елементами, що в свою чергу покращує точність рекомендацій. Вона показує кращу ефективність за традиційні методи спільної фільтрації, за рахунок нівелювання проблеми розрідженості даних та покращуючи точність за допомогою метрик (точність або ж відгук). Також автор зазначає, що модель демонструє хорошу масштабованість, ефективно працюючи з великими наборами даних.

У статті [6] пропонується рекомендаційна система SCoR, яка використовує синтетичні координати для оцінки переваг користувачів до елементів. Автор доводить, що цей підхід є ефективніший, ніж популярні методи (матричне факторизація та колаборативне фільтрування) з точки зору точності, стабільності та продуктивності. SCoR є стійкою до проблем холодного старту, є такою, що не потребує налаштування параметрів та здатна виявити спільні та унікальні характеристики користувачів та елементів. Автори зазначають, що система показує високу продуктивність на реальних даних у порівнянні з іншими методами.

3. Мета і задачі дослідження

Метою дослідження є розробка математичної моделі збору і обробки даних для рекомендаційної системи вибору дисциплін для студента з використанням комбінованих методів спільної фільтрації та матричної факторизації, що дозволяє створювати персональні навчальні плани для студентів. Основним із завдань є побудова та оптимізація алгоритму рекомендаційної системи, яка прогнозує предмети для вивчення, ґрунтуючись на попередніх результатах навчання студента та на результатах навчання інших студентів цієї ж спеціальності.

4. Результати дослідження

В ході дослідження параметрів, необхідних для надання рекомендацій щодо вибору навчальних дисциплін, було проаналізовано дані щодо освітньої діяльності студента у навчальному закладі. В основі математичної моделі рекомендаційної системи дисциплін запропоновано використати методи спільної фільтрації та матричної факторизації.

Спільна фільтрація – це поширений метод рекомендацій, на основі якого дається прогноз ґрунтуючись на вподобаннях інших користувачів із подібними інтересами. Такий прогноз

може бути на основі користувачів або об'єктів (рекомендації на основі схожих об'єктів, які оцінювали інші користувачі) [7].

Матрична факторизація – це спеціальна техніка, яка полягає в розкладі великої матриці оцінок на дві менші, із яких одна представляє собою користувачів, а інша – об'єкти, із якими взаємодіють користувачі. Такі матриці допомагають виявити приховані зв'язки між користувачами та об'єктами, що в свою чергу дозволяє передбачити оцінки і робити точніші рекомендації [8].

Головна мета цієї моделі – дати прогноз оцінки студентів для предметів, які доступні йому на вибір, опираючись на попередні оцінки студента та оцінки інших студентів тієї ж спеціальності. В основі моделі лежить матриця оцінок R , де рядки представляють студентів, а стовпці – предмети. Кожен елемент цієї матриці R_{ij} – це оцінка студента i за предмет j .

Математично задача зводиться до апроксимації матриці [9] R за допомогою двох матриць латентних факторів: одна матриця для студентів U , а інша для предметів V . Кількість латентних факторів K визначається за допомогою гіперпараметрів моделі. Прогнозування оцінок здійснюється за допомогою скалярного добутку відповідних рядків цих матриць, що в свою чергу дозволяє передбачити оцінки студентів для предметів:

$$\widehat{R}_{ij} = U_i \cdot V_j^T = \sum_{k=1}^K U_{jk} \cdot V_{jk}, \quad (1)$$

де U_j – вектор латентних факторів студента i , V_j – вектор латентних факторів предмету j , $\widehat{R}_{ij}$ – прогнозована оцінка студента i за предмет j .

Також ми будемо використовувати функцію втрат, щоб мінімізувати різницю між реальними оцінками та прогнозованими. Функція втрат – це математична функція, яка вимірює різницю між прогнозованим результатом моделі та фактичними даними. Вона використовується для оцінки точності моделі та мінімізації помилок під час навчання, зменшуючи втрати при оптимізації [10]. Цією функцією в нашому випадку буде середньоквадратична помилка (MSE) [11], яка дозволяє виміряти точність передбачення:

$$MSE = \frac{\sum_{(i,j) \in R_{train}} (R_{ij} - \widehat{R}_{ij})^2}{|R_{train}|}, \quad (2)$$

де $|R_{train}|$ – кількість відомих елементів у тренувальному наборі даних, R_{ij} – реальна оцінка студента i за предмет j .

Це дозволяє судити про якість моделі та її здатність правильно прогнозувати оцінки для предметів, які студент ще не проходив.

Розглянемо також розрахунок середньої квадратичної помилки разом із використанням норм Фробеніуса [12], яку часто використовується (приміром в праці [13], чи в роботі [14], тощо) для оцінки того, наскільки великі або малі є елементи в матриці, що важливо для регуляризації в методах, таких як матрична факторизація. Моделі, які використовують матричну факторизацію, часто мають велику кількість елементів у матрицях U і V , і якщо не обмежувати ці параметри, то це може призвести до перенавчання моделі. Норм Фробеніуса допомагає підтримувати баланс між складністю і точністю прогнозів, що дозволяє отримати стійкіші моделі, які краще працюють на нових або ж невідомих даних.

Тож ми можемо додати до функції втрат своєрідний штраф за величину елементів в матрицях латентних факторів U і V через норму Фробеніуса. Дані зміни призведуть до того, що модель буде намагатися зробити матриці U і V як можна менш великими, тим самим сприяючи більш простим і стійким до перенавчанням моделям. Така формула буде виглядати наступним чином:

$$MSE = \frac{\sum_{(i,j) \in R_{train}} (R_{ij} - \widehat{R}_{ij})^2}{|R_{train}|} + \lambda (||U||_F^2 + ||V||_F^2), \quad (3)$$

де $|R_{train}|$ – кількість відомих елементів у тренувальному наборі даних; $\|U\|_F^2 + \|V\|_F^2$ – це норма Фробеніуса для матриць U та V , яку використовуються для запобігання перенаванчання; λ – параметр регуляризації, який контролює вплив регуляризації на мінімізацію помилки, тобто наскільки сильно ми хочемо обмежити величину елементів матриць U та V . Більші значення значно зменшують величину латентних факторів, а менші – на противагу їм, дають більше свободи для цих факторів.

Для оцінки ефективності моделі ми використовуємо середньоквадратичний корінь помилки (RMSE) [15], який дає змогу наочно оцінити, наскільки точно модель відновлює відсутні оцінки. Формула для RMSE виглядає так:

$$RMSE = \sqrt{\frac{\sum_{(i,j) \in R_{train}} (R_{ij} - \widehat{R}_{ij})^2}{|R_{train}|}} \quad (4)$$

У поточній реалізації ми використовуємо ML.NET, що дозволяє автоматизувати процес факторизації матриць і мінімізації функції втрат. Використання вже готових інструментів значно спрощує розробку моделі та її налаштування. Основні етапи цього процесу включають тренування моделі, визначення параметрів тренера (кількість латентних факторів або ж кількість ітерацій), а також оптимізацію моделі через регуляризацію та градієнтний спуск. Наприклад, програмний код, який здатен дати рекомендації щодо предметів для студента та містить усі вищезгадані функції, виглядає наступним чином:

```
var mlContext = new MLContext();
List<StudentDiscipline> sameSpecialityDisciplinesScores =
    _studentDisciplineRepository.GetSameSpecialityDisciplinesScores(actualStudentId);
IDataView trainingData = mlContext.Data.LoadFromEnumerable(sameSpecialityDisciplinesScores);
DataOperationsCatalog.TrainTestData dataSplit = mlContext.Data.TrainTestSplit(trainingData, testFraction: 0.2);
IDataView trainSet = dataSplit.TrainSet;
IDataView testSet = dataSplit.TestSet;

var options = new MatrixFactorizationTrainer.Options {
    MatrixColumnIndexColumnName = "StudentIdEncoded",
    MatrixRowIndexColumnName = "CourseIdEncoded",
    LabelColumnName = "Label",
    NumberOfIterations = 50,
    ApproximationRank = 100,
    Lambda = 0.1
};

EstimatorChain<MatrixFactorizationPredictionTransformer> pipeline = mlContext.Transforms.Conversion
    .MapValueToKey(outputColumnName: "StudentIdEncoded", inputColumnName: "StudentId")
    .Append(mlContext.Transforms.Conversion.MapValueToKey(outputColumnName: "CourseIdEncoded", inputColumnName:
        "CourseId"))
    .Append(mlContext.Transforms.Text.FeaturizeText(outputColumnName: "CourseTypeFeaturized", inputColumnName:
        "CourseType"))
    .Append(mlContext.Recommendation().Trainers.MatrixFactorization(options));
TransformerChain<MatrixFactorizationPredictionTransformer> model = pipeline.Fit(trainSet);
IDataView predictions = model.Transform(testSet);
RegressionMetrics metrics = mlContext.Regression.Evaluate(predictions, labelColumnName: "Label",
    scoreColumnName: "Score");
Console.WriteLine($"Root Mean Squared Error: {metrics.RootMeanSquaredError}");
PredictionEngine<StudentDiscipline, CourseRecommendation> predictionEngine =
    mlContext.Model.CreatePredictionEngine<StudentDiscipline, CourseRecommendation>(model);
List<Discipline> notSelectedDisciplines = GetNotPassedNearDisciplines(actualStudentId);
IOrderedEnumerable<KeyValuePair<int, float>> predictionsForNewSubjects = GetPredictionsForNewSubjects
    (predictionEngine, notSelectedDisciplines, studentIdsForCalculation.First());
return predictionsForNewSubjects;
```

Рис.1. Модель рекомендаційної системи, яка реалізована за допомогою інструментів ML.NET

Де матрична факторизація реалізується через налаштування параметрів для тренування моделі, перетворення ідентифікаторів студентів і курсів у числові значення, а потім застосування методу матричної факторизації до оброблених даних, щоб створити модель, яка передбачає, які курси можуть бути цікаві студенту:

```
var options = new MatrixFactorizationTrainer.Options {
    MatrixColumnIndexColumnName = "StudentIdEncoded",
    MatrixRowIndexColumnName = "CourseIdEncoded",
    LabelColumnName = "Label",
    NumberOfIterations = 50,
    ApproximationRank = 100,
    Lambda = 0.1
};
EstimatorChain<MatrixFactorizationPredictionTransformer> pipeline = mlContext.Transforms.Conversion
    .MapValueToKey(outputColumnName: "StudentIdEncoded", inputColumnName: "StudentId")
    .Append(mlContext.Transforms.Conversion.MapValueToKey(outputColumnName: "CourseIdEncoded", inputColumnName:
        "CourseId"))
    .Append(mlContext.Transforms.Text.FeaturizeText(outputColumnName: "CourseTypeFeaturized", inputColumnName:
        "CourseType"))
    .Append(mlContext.Recommendation().Trainers.MatrixFactorization(options));
TransformerChain<MatrixFactorizationPredictionTransformer> model = pipeline.Fit(trainSet);
```

Рис.2. Калькуляція матричної факторизації за допомогою інструментів ML.NET

Норма Фробеніуса в ML.NET вказується в параметрах для тренування моделі та позначається параметром *Lambda* [16]. Значення 0.1 дозволяє моделі більш вільно підлаштовуватися під дані, що може призвести до перенавчання, якщо модель занадто сильно підлаштовується під специфічні особливості навчальних даних:

```
var options = new MatrixFactorizationTrainer.Options {
    MatrixColumnIndexColumnName = "StudentIdEncoded",
    MatrixRowIndexColumnName = "CourseIdEncoded",
    LabelColumnName = "Label",
    NumberOfIterations = 50,
    ApproximationRank = 100,
    Lambda = 0.1
};
```

Рис.3. Норма Фробеніус для матричної факторизації в ML.NET

Бібліотека ML.NET дозволяє автоматично розрахувати наскільки точно модель відновлює відсутні оцінки, за допомогою розрахунку кореня середньоквадратичної помилки:

```
RegressionMetrics metrics = mlContext.Regression.Evaluate(predictions, labelColumnName: "Label",
    scoreColumnName: "Score");
Console.WriteLine($"Root Mean Squared Error: {metrics.RootMeanSquaredError}");
```

Рис.4. Калькуляція кореня середньоквадратичної помилки за допомогою інструментів ML.NET

Система розраховує корінь середньоквадратичної помилки як під час розрахунків матричної факторизації, так і дозволяє побачити середнє значення цієї величини, після всіх розрахунків:

iter	tr_rmse	obj
0	31.8862	5.1667e+06
1	16.2569	1.3436e+06
2	15.2151	1.1770e+06
3	14.4781	1.0658e+06
4	14.0306	1.0010e+06
5	13.6917	9.5328e+05
6	13.4439	9.1912e+05
7	13.2003	8.8616e+05
8	12.9829	8.5724e+05
9	12.7830	8.3108e+05
Root Mean Squared Error:		15.35790628165678

Рис.5. Приклад розрахунку кореня середньоквадратичної помилки за допомогою інструментів ML.NET

Змінюючи параметри для тренування моделі на основі матричної факторизації, можна отримати більш точні рекомендації. Чим менші значення кореня середньоквадратичної помилки (RMSE), тим вища точність передбачення моделі, оскільки RMSE є мірою відхилення між фактичними і прогнозованими значеннями, і її зменшення свідчить про покращення точності моделі. Бажаного результату можна досягти або різними способами тренування моделі, або підбором значень для властивостей *NumberOfIterations*, *ApproximationRank*, *Lambda*, які можуть варіюватися від прикладної тематики, яка вирішується, або від необхідності точності передбачення. Для даного прикладу ми обрали середні значення, які є достатніми для оптимізації ваг, апроксимації та регуляризації. Значення 15.35, яке ми отримали в ході розрахунку рекомендації, можна вважати хорошим результатом, враховуючи те, що значення передбачення рекомендаційної системи лежать в межах від 0 до 100. Однак, в статті представлений конкретний приклад та отримані значення можуть варіюватися, в залежності від початкових даних моделі. Тому даний алгоритм потребує подальшого опрацювання та покращення, не зважаючи на те, що він здатен вже надавати рекомендації для студента такого типу:

Загальноуніверситетський вибір навчальних дисциплін за уподобанням

1. Ораторське мистецтво (рекомендується до вивчення на 99.07%)	Терм. навчання: 2025-2026 Семестр: 2 Кредити: 4	Викладачі: Майданчик І.З.	☐
2. Власна справа: створення і запуск власного Інтернет-магазину (рекомендується до вивчення на 97.81%)	Терм. навчання: 2025-2026 Семестр: 2 Кредити: 4	Викладачі: Шушкова І.Ф.	☐
3. Іміджологія (рекомендується до вивчення на 97.28%)	Терм. навчання: 2025-2026 Семестр: 2 Кредити: 4	Викладачі: Сидоренко І.Г. Майданчик І.З.	☐
4. Біржова діяльність (рекомендується до вивчення на ...)	Терм. навчання: 2025-2026 Семестр: 2 Кредити: 4	Викладачі: Яворська В.О.	☐

Вибрано кредитів: 8 з 8

Рис.6. Приклад рекомендації дисциплін для студента

5. Висновки і перспективи подальших досліджень

У результаті проведеного дослідження було обґрунтовано використання математичної моделі на основі матричної факторизації та мінімізації функції втрат для рекомендації вибору предметів студентами. Базовими параметрами для побудови моделі є попередні результати навчання та оцінки студентів тієї ж спеціальності за останні три роки з відповідних дисциплін. Запропонована модель була апробована та продемонструвала свою ефективність у прогнозуванні оцінок для предметів, що студенту пропонуються на вибір.

За допомогою використання інструментів ML.NET процес матричної факторизації і оптимізації моделі був автоматизований, що значно спростило реалізацію та налаштування системи. В ході дослідження встановлено як застосування середньоквадратичної помилки для оцінки точності прогнозів дозволяє досягти оптимізації результатів і зменшення різниці між реальними та передбаченими оцінками.

Розроблена математична модель для рекомендаційної системи вибору навчальних дисциплін має значний потенціал для використання в освітніх установах з метою підвищення ефективності навчання та надання студентам більш точних та адаптованих під конкретного студента рекомендацій. Застосування подібних систем забезпечує персоналізований підхід до навчання і підвищення якості освіти.

Проте, варто зазначити, що є кілька напрямків для подальшого вдосконалення моделі. Наприклад, можна інтегрувати контентну фільтрацію – метод, який здатен робити пропозиції для користувачів, на основі їх характеристик та вподобань [17]. В нашій ситуації, такий тип алгоритму враховував би наприклад специфікації предметів, такі як тип, складність, викладач і т.д., що дозволить поліпшити точність рекомендацій, зменшуючи залежність від фактору

оцінок студентів. Це можна реалізувати через комбінування латентних факторів V_j з контентними характеристиками предметів C_j :

$$\widehat{R}_{ij} = (U_i) \cdot (V_j + C_j)^T \quad (5)$$

Ще одним покращенням може бути обчислення схожості між предметами або студентами. Наприклад, схожість між предметами можна обчислювати через скалярний добуток латентних векторів предметів [18]:

$$im(j, j') = V_j \cdot \frac{V_{j'}^T}{\|V_j\|_2 \|V_{j'}\|_2} \quad (6)$$

Це дозволить на основі цієї схожості генерувати більш персоналізовані рекомендації. Крім того, можна комбінувати матричну факторизацію з іншими методами, такими як глибоке навчання або додаткові моделі, що будуть використовувати інформацію про навчання студента.

Список використаної літератури

1. Hyeyoung Ko, Suyeon Lee, Yoonseo Park, Anna Choi. (2022). A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. [Електронний ресурс]. – Режим доступу: <https://www.mdpi.com/2079-9292/11/1/141>
2. Кюльян Артур, Польгуль Тетяна, Хазін Марат. Математична модель рекомендаційного сервісу на основі методу колаборативної фільтрації. [Електронний ресурс]. – Режим доступу: <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/7911/226-227.pdf?sequence=1&isAllowed=y>
3. Yimin Cui. (2021). Intelligent Recommendation System Based on Mathematical Modeling in Personalized Data Mining. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/349701292_Intelligent_Recommendation_System_Based_on_Mathematical_Modeling_in_Personalized_Data_Mining
4. Filali Zegzouti S., Banouar O., Benslimane M. (2024). Recommendation systems techniques based on generative models and matrix factorization: a survey. [Електронний ресурс]. – Режим доступу: <https://science.lpnu.ua/mmc/all-volumes-and-issues/volume-11-number-4-2024/recommendation-systems-techniques-based>
5. Aamir Fareed, Saima Hassan, Samir Brahim Belhaouari, Zahid Halim. (2023). A collaborative filtering recommendation framework utilizing social networks. [Електронний ресурс]. – Режим доступу: <https://www.sciencedirect.com/science/article/pii/S2666827023000488>
6. Harris Papadakis, Costas Panagiotakis, Paraskevi Fragopoulou. (2017). SCoR: A Synthetic Coordinate based Recommender System. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/313835321_SCoR_A_Synthetic_Coordinate_based_Recommender_System
7. Collaborative Filtering in Recommender System: An Overview. (2023). [Електронний ресурс]. – Режим доступу: <https://medium.com/@evelyn.eve.9512/collaborative-filtering-in-recommender-system-an-overview-38dfa8462b61>
8. Matrix factorization. [Електронний ресурс]. – Режим доступу: <https://developers.google.com/machine-learning/recommendation/collaborative/matrix>
9. Approximated Matrix. [Електронний ресурс]. – Режим доступу: <https://www.sciencedirect.com/topics/computer-science/approximated-matrix>
10. Dave Bergmann, Cole Stryker. (2024). What is a loss function? [Електронний ресурс]. – Режим доступу: <https://www.ibm.com/think/topics/loss-function>
11. Mean-Squared-Error. [Електронний ресурс]. – Режим доступу: <https://www.sciencedirect.com/topics/engineering/mean-squared-error>
12. Frobenius Norm. [Електронний ресурс]. – Режим доступу: <https://www.sciencedirect.com/topics/mathematics/frobenius-norm>

13. Н. А. Яремчук, О. Ю. Редьбога, О. М. Сікоза. Особливості арифметизації дискретних вербальних шкал. [Електронний ресурс]. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/8c59d2a8-4bc1-4caa-91df-39024f503020/content>
14. Roland Herzog, Frederik Köhne, Leonie Kreis, Anton Schiela. (2023). Frobenius-type norms and inner products of matrices and linear maps with applications to neural network training. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/2311.15419>
15. Dhruval Patel, Foram Patel, Uttam Chauhan. (2023). Recommendation Systems: Types, Applications, and Challenges. [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/370498330_Recommendation_Systems_Types_Applications_and_Challenges
16. MatrixFactorizationTrainer.Options.Lambda Field. [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/uk-ua/dotnet/api/microsoft.ml.trainers.matrixfactorizationtrainer.options.lambda?view=ml-dotnet-preview>
17. Jacob Murel, Eda Kavlakoglu. (2024). What is content-based filtering? [Електронний ресурс]. – Режим доступу: <https://www.ibm.com/think/topics/content-based-filtering>
18. Robert Sheldon. (2022). Dot product (scalar product). [Електронний ресурс]. – Режим доступу: <https://www.techtarget.com/whatis/definition/dot-product-scalar-product>