

УДК 004.8.032.26

## ПІДВИЩЕННЯ КОЕФІЦІЄНТА КОНВЕРСІЇ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ: КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ

Конотопчик А.М., студент, [artemkonotopchik2003@gmail.com](mailto:artemkonotopchik2003@gmail.com), ЛНТУ

Мельник К.В., к. т. н., доцент, [ekaterinamelnik@gmail.com](mailto:ekaterinamelnik@gmail.com), ЛНТУ

Лавренчук С.В., к. т. н., доцент, [LavrSveet@gmail.com](mailto:LavrSveet@gmail.com), ЛНТУ

В умовах великої кількості даних та очікуваного зростання їх обсягу, застосування глибокого навчання для виявлення цінних знань стає інноваційним рішенням. Цей підхід може бути використаний для створення рекомендаційних систем, що сприяють підвищенню коефіцієнту конверсії.

При стрімкому зростанні кількості онлайн покупок збільшення коефіцієнту конверсії стає основним завданням онлайн-платформ, які бажають збільшити кількість транзакцій на своїх платформах. Системи CRO (Conversion Rate Optimization) направлені на те, щоб якомога більше користувачів виконували бажані дії, такі як: створення облікового запису, запитування контактів або здійснення покупки [1].

Класичними методами оптимізації рейтингу конверсій є A/B тестування [2], веб-аналітика та персоналізація контенту [3]. Дані методики є вживаними та ефективними, однак їх класичне використання не дозволяє обробляти велику кількість даних, або швидко генерувати результати. Задля оптимізації даних рішень в даній статі розглядається використання машинного навчання.

Для збільшення CRO необхідно передбачити можливі бажання клієнта, або, зацікавити його в здійсненні певних дій. У випадку онлайн-магазину такими діями можна вважати: перегляд інформації про товар, залишення відгуку чи коментарів, здійснення покупки. Задля здійснення даної дії необхідна рекомендаційна система, яка здатна передбачити можливі вподобання користувача. Одним з методів реалізації рекомендаційної системи для збільшення коефіцієнту конверсії є колаборативна фільтрація.

Колобаративна фільтрація використовує інформацію про інших користувачів або елементів, з якими вони взаємодіють, задля встановлення взаємозв'язків між ними [4].

В даному дослідженні розглядається методика колаборативної фільтрації на основі нейронної мережі, що розраховує взаємозв'язки між оцінками користувачів до конкретних товарів. Для дослідження використовувався датасет «Amazon – Ratings (Beauty Products)» [5]. Датасет містить понад 2 мільйони оцінок користувачів на понад 249 тисяч товарів і включає наступну інформацію: ідентифікатор користувача, ідентифікатор товару, рейтингова оцінка (в діапазоні [1:5] на основі задоволеності клієнтів), часову позначку рейтингу.

Зважаючи на конфігурацію даних в датасеті була обрана глибока нейронна мережа прямого поширення. Дана мережа містить 72216450 тренувальних параметрів, складається з вхідних, двох прихованих та одного вихідного шару (рисунок 1). Вхідними шарами є пари «користувач-продукт», а вихідними даними є рейтинг.

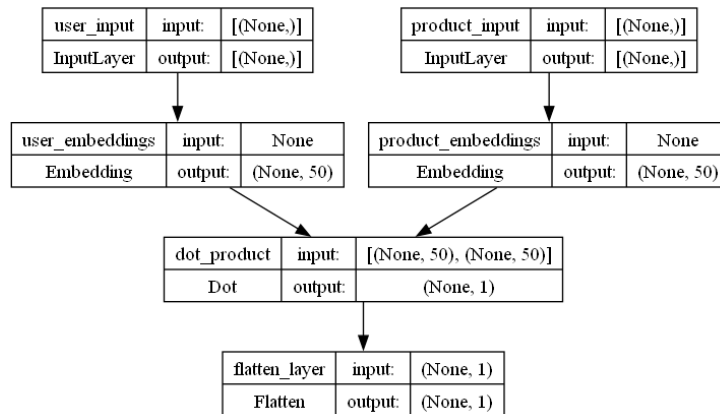


Рисунок 1 – Схема архітектури нейронної мережі створена функцією plot\_model з бібліотеки keras

На рисунку 2 представлена програмна реалізація глибокої нейронної мережі.

```
# Функція для побудови моделі
def build_model(num_users, num_products, embedding_size):
    user_input = Input(shape=(), name='user_input')
    product_input = Input(shape=(), name='product_input')

    user_embedding = Embedding(input_dim=num_users, output_dim=embedding_size,
name='user_embeddings')(user_input)
    product_embedding = Embedding(input_dim=num_products,
output_dim=embedding_size, name='product_embeddings')(product_input)

    dot_product = Dot(axes=1, name='dot_product')([user_embedding,
product_embedding])
    flatten = Flatten(name='flatten_layer')(dot_product)

    model = Model(inputs=[user_input, product_input], outputs=flatten)
    return model
```

Рисунок 2 – Програмна реалізація архітектури нейронної мережі

Навчальна вибірка складається з 2000000 оцінок та розподілена на навчальну та тестову частину у співвідношенні 9:1, валідаційна вибірка становить 10% від тестової вибірки. Гіперпараметри навчання мережі наведені у таблиці 1.

Таблиця 1 – Гіперпараметри при навчанні нейронної мережі

Гіперпараметр	Опис
Кількість прихованих шарів	2
Кількість епох	10
Валідаційна вибірка	0.1 (10%)
Оптимізатор	Adam
Розмір міні-вибірки	128
Функція визначення втрат	Середньоквадратичне відхилення

Після навчання, для перевірки моделі, проведено оцінку точності на тестовій вибірці, отримавши значення середньоквадратичного відхилення та середнього абсолютного відхилення.

Значення похибок на тестових даних представлені на рисунку 3.

Test Metrics: {'mean\_squared\_error': 0.6940356492996216, 'mean\_absolute\_error': 0.7537084221839905}

Рисунок 3 – Значення похибок на тестовому наборі

Створена глибока нейронна мережа, що дозволяє передбачати можливі оцінки користувача на основі попередніх виставлених оцінок. Побудована функція формування рейтингу продуктів для конкретного користувача, яка використовує навчену нейронну мережу (рисунок 4). Результат виконання даної функції представлений на рисунку 5.

```
# Функція для рекомендації продуктів для конкретного користувача
def recommend_product_for_user(user_id):
    user_ids = [user_id] * len(allProducts)
    data = pd.DataFrame({'UserId': user_ids, 'ProductId': allProducts})

    data = data[['UserId', 'ProductId']]

    # Передбачення рейтингу для користувача та всіх товарів
    predicted_ratings = model.predict([data['UserId'], data['ProductId']])
    predicted_ratings = scaler.inverse_transform(np.array(predicted_ratings))

    # Фрейм з передбаченими оцінками
    all_products_per_user = pd.DataFrame({
        'UserId': user_ids,
        'ProductId': allProducts,
        'Rating': predicted_ratings.flatten() # Flatten the predictions if
        necessary
    })

    all_products_per_user_sorted =
all_products_per_user.sort_values(by='Rating', ascending=False)
    return all_products_per_user_sorted
```

Рисунок 4 – Функція формування передбачень рейтингу товарів для конкретного користувача

```
# Приклад виклику функції  
data_per_user = recommend_product_for_user(45)  
print(data_per_user)
```

	UserId	ProductId	Rating
	91390	45	91390 1.931937
	85555	45	85555 1.915688
	243455	45	243455 1.902118
	41514	45	41514 1.830711
	59527	45	59527 1.814345
	...	...	...
	87958	45	87958 0.092461
	113179	45	113179 0.082648
	12233	45	12233 0.060229
	144825	45	144825 0.036003
	230662	45	230662 -0.071776

[243497 rows x 3 columns]

Рисунок 5 – Результат виконання функції формування передбачень рейтингу товарів

**Висновок.** В даній статі були використанні методи машинного навчання, а саме глибокі нейронні мережі, для оптимізації рейтингу конверсії на основі алгоритму колобаративної фільтрації. Показані переваги використання нейронних мереж при роботі з великими даними, продемонстровано швидке генерування оцінок продуктів (більш ніж 240 тисяч одиниць). Це доводить, що рекомендаційні системи на основі нейронних мереж є ефективним інструментом для проведення оптимізації рейтингу конверсії.

#### Список використаних джерел

1. Ascend by Evolv: AI-Based Massively Multivariate Conversion Rate Optimization / R. Miikulainen та ін. 2020. URL: <https://doi.org/10.1609/aimag.v41i1.5256> (дата звернення: 19.02.2024).
2. A/B Testing: A Systematic Literature Review / Federico Quin та ін. 2023. URL: <https://arxiv.org/pdf/2308.04929.pdf> (дата звернення 19.02.2024)
3. Digital Marketing Strategy to Increase Sales Conversion on E-commerce Platforms / Yudianto Joko Purnomo. 2023. URL: <https://doi.org/10.61100/adman.v1i2.23> (дата звернення 20.02.2024)
4. Recommender Systems Based on Collaborative Filtering Using Review Texts— A Survey / M. Srifi та ін. 2020. URL: <https://doi.org/10.3390/info11060317> (дата звернення: 20.02.2024).
5. Amazon - Ratings (Beauty Products). Kaggle: Your Machine Learning and Data Science Community. URL: <https://www.kaggle.com/datasets/skillsmugger/amazon-ratings> (дата звернення: 21.02.2024).