

ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПЕРСОНАЛИЗАЦИИ РЕКОМЕНДАЦИЙ В ФУДТЕХ ИНДУСТРИИ

Кукитз П.В.

ООО «Рестомания», Москва, Россия

Ключевые слова: персонализация в фудтехе, разработка рекомендательных систем, машинное обучение для персонализации, коллаборативная фильтрация, контент-based filtering, кластеризация, матричная факторизация, Reinforcement Learning, глубокое обучение.

Аннотация. Статья посвящена исследованию вопросов применения алгоритмов машинного обучения для персонализации рекомендаций в фудтех-индустрии. В работе анализируются различные алгоритмы, используемые для анализа пользовательских предпочтений и формирования персонализированных предложений (коллаборативная фильтрация, контент-based filtering, кластеризация, матричная факторизация, Reinforcement Learning и глубокое обучение). Описываются преимущества и недостатки каждого алгоритма, приводятся примеры их реализации на языке Python. Затрагиваются этические вопросы, связанные с использованием данных алгоритмов, которые необходимо учитывать разработчикам. По итогам проведенного исследования отмечается, что применение машинного обучения для персонализации рекомендаций в фудтех-индустрии является перспективным направлением, позволяющим повысить качество предоставляемых услуг и удовлетворенность пользователей.

APPLICATION OF MACHINE LEARNING FOR PERSONALIZING RECOMMENDATIONS IN THE FOOD TECH INDUSTRY

Kukitz P. V.

LLC Restomania, Moscow, Russia

Keywords: personalization in food technology, development of recommendation systems, machine learning for personalization, collaborative filtering, content-based filtering, clustering, matrix factorization, Reinforcement Learning, deep learning.

Abstract. This article explores the application of machine learning algorithms for personalizing recommendations in the food tech industry. The study analyzes various algorithms used for analyzing user preferences and generating personalized recommendations (collaborative filtering, content-based filtering, clustering, matrix factorization, Reinforcement Learning, and deep learning). The advantages and disadvantages of each algorithm are described, and examples of their implementation in Python are provided. Ethical issues related to the use of these algorithms, which developers need to consider, are also discussed. The study concludes that the application of machine learning for personalizing recommendations in the food tech industry is a promising direction that can enhance the quality of services provided and user satisfaction.

Стремительные темпы цифровизации бизнеса определяют роль цифровых технологий как инструмента повышения качества исполнения тех или иных функций в деятельности организации. Вместе с тем, сегодня многие из вопросов цифровизации и применения цифровых решений остаются разрозненными, требуют прикладного освещения, связанного с уточнением инструментов и алгоритмов их технического (программного) построения. С точки зрения разработки рекомендательных систем и применения алгоритмов, направленных на персонализацию предложений, возникает особая

перспектива раскрытия применения машинного обучения и его алгоритмов, что становится фундаментом на пути к последующей разработке эффективного инструмента для исполнения системных задач.

Актуальность исследования вопросов применения машинного обучения для персонализации рекомендаций в фудтех индустрии связана с тем, что фудтех индустрия характеризуется специфическим набором данных, которые используются в типовых алгоритмах, применяющихся для решения поставленной задачи. В частности, в практике разработки программного обеспечения особую роль приобретают данные, которые отражают общие пользовательские предпочтения, возможные аллергические реакции на продукты, выбор конкретных продуктов в пищу, историю заказов, просмотры, занесение определенных блюд или ресторанов в избранное (и т.п.), а также данные, которые компания получает в контексте определенных действий, начиная от времени суток, местоположения, назначения заказов и др. Перечисленные данные в дальнейшем формируют выборку для применения алгоритмов машинного обучения с последующим формированием персонализированных рекомендаций, в связи с чем требуется их полноценное установление и определение перспективных направлений применения в алгоритмах машинного обучения для персонализации рекомендаций в фудтех индустрии.

Цель исследования – описать применение алгоритмов машинного обучения для анализа предпочтений пользователей и формирования персонализированных предложений.

Исследованием применения машинного обучения в задачах персонализации предложений занимались многие авторы, например Е.П. Попова, В.Н. Леоненко, Д.А. Сапрыкин, К.В. Подольский, А.В. Панов, А.Н. Черняков, М.Ш. Дибиров и другие. Исследования перечисленных авторов показывают [1-3], что применение алгоритмов машинного обучения в решении задач определения предпочтений пользователей с последующей персонализацией предложений обеспечивает не только более эффективное и качественное формирование этих рекомендаций, с развитием пользовательского опыта и повышением эффективности бизнеса, но и позволяет создавать прикладные постоянно совершенствующиеся алгоритмы. Авторами резюмируется система требований к таким программным системам, их архитектуре, с формированием возможности масштабирования, улучшения, работоспособностью в пиковые нагрузки и постоянной оптимизацией. Существование множества алгоритмов машинного обучения предполагает определение наиболее жизнеспособных в контексте реализуемых задач, в целях обеспечения точности, учета специфических особенностей применяемых данных, а также последующей интеграцией с внешним приложением и интерфейсами, например, с созданием API. Все обозначенные задачи находят свое отражение в деятельности разработчиков фудтех индустрии и связываются с повышением качества и работоспособности программных алгоритмов.

Так, различные алгоритмы находят собственное применение в решении задач разработки рекомендательных сервисов; например, исследование Т.В. Смоленчук показывает, что применение метода коллаборативной фильтрации для рекомендательных сервисов обеспечивает должный уровень реализации программных задач, которые планирует достичь команда разработчиков при подготовке алгоритмов машинного обучения [4]. На практике применение коллаборативной фильтрации сводится к использованию User-based и Item-based методов, которые позволяют рассматривать схожесть пользователей и схожесть объектов соответственно в качестве основания для предоставления рекомендаций. Основой для коллаборативной фильтрации является матрица предпочтений (ratings matrix), в которой строки соответствуют пользователям, а столбцы – объектам (например, блюда или типы ресторанов). Элементами матрицы выступают оценки, которые пользователи присвоили объектам (табл. 1).

Табл. 1. Матрица оценки ресторанов пользователями (составлено автором)

	Ресторан А	Ресторан В	Ресторан С
Пользователь А	5	3	?
Пользователь В	5	?	4
Пользователь С	?	4	3

Отметим, что в предложенном примере матрицы «?» соответствует отсутствию или неизвестности оценки, которую пользователь поставил ресторану. Для реализации коллаборативной фильтрации и проведения оценок схожести предпочтений пользователей (или оцениваемых объектов) используются метрики косинусного сходства, критерий корреляции Пирсона или коэффициент Жаккара, в зависимости от задачи и предпочтений. Например, косинусное сходство имеет следующий вид:

$$\text{similarity}(A, B) = \cos(\theta) = (A \cdot B) / (\|A\| \|B\|).$$

С точки зрения программирования, представим пример реализации коллаборативной фильтрации на библиотеке Surprise на языке Python (рис. 1).

На рисунке 1 проиллюстрировано применение user-based коллаборативной фильтрации. Dataset.load_builtin('ml-100k') позволяет быстро начать работу с библиотекой, использовать при этом известный набор данных для тестирования, а включение sim_options позволяет явно указать метрику сходства (cosine) и тип фильтрации (user_based).

В представленном примере и описании применения алгоритма коллаборативной фильтрации на библиотеке Surprise на языке Python нами были упомянуты известные данные; отметим, что задача сбора информации для рекомендательных систем в целом является одной из концептуально значимых. В фудтех индустрии в выборку данных включаются следующие типы данных (рис. 2).

```

from surprise import Dataset, Reader, KNNBasic
from surprise.model_selection import train_test_split
from surprise import accuracy

# Загрузка данных
data = Dataset.load_builtin('ml-100k')
trainset, testset = train_test_split(data, test_size=0.25)

# User-based коллаборативная фильтрация
sim_options = {'name': 'cosine', 'user_based': True}
algo = KNNBasic(sim_options=sim_options)

# Обучение модели
algo.fit(trainset)

# Прогнозирование
predictions = algo.test(testset)

# Оценка точности
print(accuracy.rmse(predictions))

```

Рис. 1. Применение алгоритма коллаборативной фильтрации на библиотеке Surprise на языке Python (составлено автором)

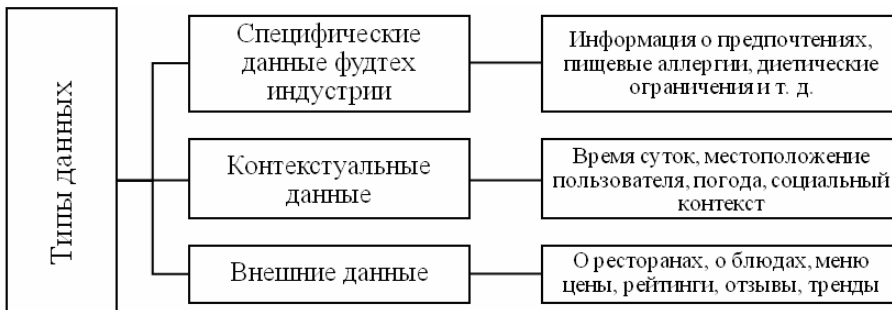


Рис. 2. Типы данных для применения алгоритмов машинного обучения и анализа предпочтений пользователей и формирования персонализированных предложений (составлено автором)

Д.Б. Инамова и М.А. Чиркина, выражая схожие воззрения касаясь задач сбора информации для рекомендательных систем, выделяют не менее значимый алгоритм машинного обучения – content-based filtering, который позволяет подбирать схожий контент для пользователя на основе выставляемых им оценок и формировать в соответствии с ними необходимые рекомендации [5]. Применение данного алгоритма предполагает учет характеристик объектов и предпочтений пользователей, с рекомендацией тех объектов, которые похожи на предпочтения (понравившиеся пользователю), с учетом свойств или иных характеристик. Для реализации контент фильтрации

реализуется функция анализа атрибутов объектов, а также изучается профиль пользователя (его предпочтения). Объекты и пользовательские профили представляются в виде векторов, где каждая компонента вектора соответствует определенному атрибуту, например:

Ресторан А: [1, 1, 0] # категории: [Итальянская, Китайская, Индийская];

Ресторан В: [0, 0, 1] # категории: [Итальянская, Китайская, Индийская];

Ресторан С: [1, 0, 1] # категории: [Итальянская, Китайская, Индийская];

User Profile: [0.7, 0.3, 0.5] # предпочтения пользователя по категориям: [Итальянская, Китайская, Индийская].

```
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity

# Векторы ресторанов
restaurant_a = np.array([1, 1, 0])
restaurant_b = np.array([0, 0, 1])
restaurant_c = np.array([1, 0, 1])

# Профиль пользователя
user_profile = np.array([0.7, 0.3, 0.5])

# Список ресторанов
restaurants = ['Restaurant A', 'Restaurant B', 'Restaurant C']
restaurant_vectors = [restaurant_a, restaurant_b, restaurant_c]

# Вычисление косинусного сходства
similarities = [cosine_similarity([user_profile], [restaurant])[0][0] for
restaurant in restaurant_vectors]

# Сортировка ресторанов по сходству
recommendations = sorted(zip(restaurants, similarities), key=lambda x:
x[1], reverse=True)

# Вывод рекомендаций
for restaurant, similarity in recommendations:
    print(f'{restaurant}: {similarity:.2f}')
```

Рис. 3. Применение контентной фильтрации для рекомендательных систем ресторана на примере библиотеки scikit-learn на Python (составлено автором)

В результате таких оценок, пользователю будет ближе всего Ресторан С, так как он предлагает кухни, которые нравятся пользователю (итальянская и индийская). Однако отметим, что представленный пример несколько упрощен, поскольку векторы признаков и профиля пользователя заданы вручную, что нереалистично для реальных систем; к тому же, он не подходит для большого количества ресторанов и пользователей, не является масштабируемым. И если упрощение модели осуществлено целенаправленно, то разрешение проблемы масштабируемости стоит особенно остро, требует хранения данных о ресторанах и пользователях в базах данных, оптимизированных для быстрого поиска и выборки данных (например, PostgreSQL с расширением pgvector для работы с векторами). Кроме того, в качестве рекомендации по улучшению такого кода стоит рассматривать

применение фреймворков для распределенных вычислений (например, Apache Spark или Dask), чтобы параллелизовать вычисления и обрабатывать большие объемы данных. Масштабируемость такого алгоритма можно обеспечить за счет использования специализированных индексов для быстрого поиска ближайших соседей в пространстве векторов (например, FAISS, Annoy или HNSW). Заметим, что на практике выбор конкретного подхода зависит от объема данных, доступных ресурсов и требуемой точности системы рекомендаций.

Не менее значимым алгоритмом машинного обучения рекомендательных систем для фудтех индустрии является алгоритм кластеризации, который применяется для проведения группирования объектов для обеспечения их схожести друг с другом. Ценность кластеризации объясняется тем, что кластеризация позволяет обеспечить выявление скрытых структур данных с их последующей классификацией по сходству, что используется при реализации процедур анализа данных, сегментации аудитории, уменьшения размерности данных и других задач. На рисунке 4 описано применение кластеризации для группировки ресторанов на основе их характеристик (тип кухни, предоставляемые услуги и т.д.) Одним из наиболее популярных методов кластеризации выступает метод K-means, рассмотрим его использование с библиотекой scikit-learn на Python (рис. 4). В результате выполнения кода формируется список ресторанов, которые группируются по кластерам. В кластер 0 включаются рестораны В и D, в кластер 1 рестораны А, С и F, в кластер 2 – ресторан Е. Метод K-means в данном случае позволяет выявлять скрытые структуры в данных и создавать кластеры, которые обладают весомым значением для последующего проведения процедур анализа и принятия решений.

В.Н. Никулин и Т.Г. Прозорова верно подчеркивают, что одним из перспективных алгоритмов машинного обучения рекомендательных систем является матричная факторизация [6], которая позволяет сформировать представление пользователей и продуктов (в нашем случае – ресторанов и блюд) в виде векторов скрытых признаков (latent factors). Алгоритм использует матрицу «пользователь-продукт» (например, рейтинги пользователей ресторанам), которая, как правило, неполная и разреженная, матрица раскладывается на две матрицы меньшей размерности: 1) Матрица «пользователь-скрытые факторы», где каждая строка представляет вектор скрытых признаков пользователя; 2) Матрица «скрытые факторы-продукт», где каждая строка представляет вектор скрытых признаков продукта. Чтобы предсказать рейтинг пользователя для определенного продукта, алгоритм перемножает соответствующие векторы пользователя и продукта. Алгоритм способен рекомендовать пользователю рестораны, основываясь на сходстве его вектора скрытых признаков с векторами ресторанов, что аналогичным образом распространяется на блюда, рекомендации по которым формируются на основе учета предпочтений пользователя и характеристики блюд в пространстве скрытых признаков. Скрытые факторы при этом находят

применение для создания персонализированных предложений, например скидок на блюда или рестораны, которые наиболее интересны пользователю.

```
import numpy as np
from sklearn.cluster import KMeans
import pandas as pd

# Данные о ресторанах
restaurants = np.array([
    [1, 1, 0], # Ресторан А: Итальянская, Китайская, Индийская
    [0, 0, 1], # Ресторан В: Итальянская, Китайская, Индийская
    [1, 0, 1], # Ресторан С: Итальянская, Китайская, Индийская
    [0, 1, 1], # Ресторан D: Итальянская, Китайская, Индийская
    [1, 1, 1], # Ресторан Е: Итальянская, Китайская, Индийская
    [1, 0, 0] # Ресторан F: Итальянская, Китайская, Индийская
])

# Имена ресторанов
restaurant_names = ['Restaurant A', 'Restaurant B', 'Restaurant C',
                    'Restaurant D', 'Restaurant E', 'Restaurant F']

# Количество кластеров (можно экспериментировать с этим числом)
num_clusters = 3

# Модель K-means
kmeans = KMeans(n_clusters=num_clusters, random_state=0)
kmeans.fit(restaurants)

# Метки кластеров
labels = kmeans.labels_

# Создание DataFrame для удобства анализа
df = pd.DataFrame({
    'Restaurant': restaurant_names,
    'Cluster': labels
})

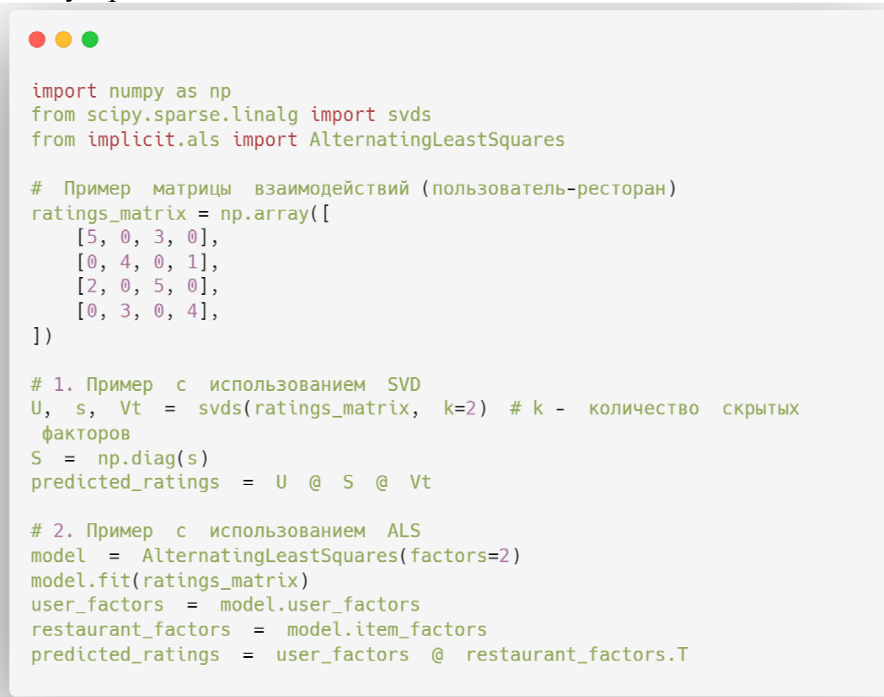
# Группировка ресторанов по кластерам
clustered_restaurants = df.groupby('Cluster')

# Вывод результатов
for cluster, group in clustered_restaurants:
    print(f"Cluster {cluster}:")
    print(group[['Restaurant']])
    print()
```

Рис. 4. Кластеризация ресторанов на основе метода K-means в библиотеке scikit-learn на Python (составлено автором)

Ключевыми техническими примерами матричной факторизации являются SVD и ALS, которые используются для разложения матрицы на три матрицы, одна из которых содержит сингулярные числа, отражающие важность каждого скрытого фактора и для приближения исходной матрицы, минимизации суммы квадратов ошибок соответственно. Программный пример реализации алгоритма представлен на рис. 5. В представленном примере используется `scipy.sparse.linalg.svds`, функция из библиотеки SciPy, которая выполняет сингулярное разложение матрицы (SVD). В коде U , s , $Vt =$

`=svds(ratings_matrix, k=2)` применяется SVD к матрице взаимодействий `ratings_matrix`. `k=2` указывает, что алгоритм направлен на извлечение 2 скрытых факторов, а результатом являются матрицы `U`, `s`, `Vt`. Для получения предсказанной матрицы рейтингов матрицы перемножаются (`predicted_ratings = U @ S @ Vt` (где `S` - диагональная матрица, полученная из `s`)). Вторая библиотека – `implicit.als`, которая специализируется на алгоритмах коллаборативной фильтрации, в том числе на методе чередующихся наименьших квадратов (ALS). В коде `model = AlternatingLeastSquares(factors=2)` создается модель ALS с двумя скрытыми факторами. `model.fit(ratings_matrix)` обучает модель на матрице взаимодействий. После обучения `model.user_factors` и `model.item_factors` включают в себя матрицы скрытых факторов для пользователей и ресторанов соответственно. `predicted_ratings = user_factors @ restaurant_factors.T` вычисляет предсказанную матрицу рейтингов путем перемножения матриц скрытых факторов. Данный код демонстрирует два разных способа выполнения матричной факторизации для предсказания рейтингов в рекомендательной системе для ресторанов, а именно SVD и ALS. Оба метода позволяют получить векторы скрытых факторов для пользователей и ресторанов, которые могут быть использованы для формирования рекомендаций, кластеризации пользователей и ресторанов, и других задач в фудтех индустрии.



```

import numpy as np
from scipy.sparse.linalg import svds
from implicit.als import AlternatingLeastSquares

# Пример матрицы взаимодействий (пользователь-ресторан)
ratings_matrix = np.array([
    [5, 0, 3, 0],
    [0, 4, 0, 1],
    [2, 0, 5, 0],
    [0, 3, 0, 4],
])

# 1. Пример с использованием SVD
U, s, Vt = svds(ratings_matrix, k=2) # k - количество скрытых факторов
S = np.diag(s)
predicted_ratings = U @ S @ Vt

# 2. Пример с использованием ALS
model = AlternatingLeastSquares(factors=2)
model.fit(ratings_matrix)
user_factors = model.user_factors
restaurant_factors = model.item_factors
predicted_ratings = user_factors @ restaurant_factors.T

```

Рис. 5. Матричная факторизация в фудтех индустрии на Python (составлено автором)

Более сложным в решении задач создания рекомендательных систем в фудтех индустрии является алгоритм Reinforcement Learning, который

фокусируется на обучении агентов, действующих в окружающей среде. Агент учится, пробует разные действия и наблюдает за получаемыми наградами или наказаниями. RL отличается от supervised learning, где алгоритм обучается на размеченных данных, и unsupervised learning, где алгоритм ищет паттерны в неразмеченных данных. Как правило, такой алгоритм применяется для персонализации внутренних процессов (оптимизация цен, логистики, запасов); однако нередко находит применение для персонализации меню, с учетом динамики пользовательских предпочтений и предложения индивидуальных решений для каждого пользователя. RL-агенты могут находить оптимальные решения для сложных задач, с которыми не справляются традиционные алгоритмы. Аналогичной перспективностью обладают алгоритмы глубокого обучения (Deep Learning, DL), которые основаны на использовании нейросетей с несколькими слоями. Наиболее перспективной областью использования глубокого обучения являются задачи персонализации на основе сложной системы вкусов и предпочтений потребителя. Такие алгоритмы достаточно точны, автоматизированы и легко подстраиваются под изменения, однако их разработка достаточно затратна, сложна, требует огромных массивов данных и вычислительных мощностей для реализации.

Так, подводя итоги обобщенному описанию существующих алгоритмов машинного обучения для создания рекомендательных систем в фудтех индустрии, сгруппируем их достоинства и недостатки в форме таблицы 2.

Кроме того, рассматривая вопросы применения обозначенных алгоритмов, важно подчеркнуть необходимость учета этических аспектов работы; в частности, существует несколько ключевых проблем этического характера, которые проявляются при использовании данных алгоритмов. Как пишут Ю.С. Харитонова, В.С. Савина и Ф. Паньини, современные машинные алгоритмы демонстрируют целую систему проблем этического характера, которые должны учитываться современными разработчиками и приниматься во внимание в качестве фундамента деятельности разработчика. Обобщая воззрения авторов, конкретизируем некоторые проблемы этики, характерные для применения рекомендательных систем в фудтех индустрии [7]. Первая проблема – возможная дискриминация со стороны системы, при которой пользователю будут предлагаться только уже опробованные блюда, рестораны, кухни, что значительно ограничивает его предпочтения, формирует предвзятое отношение к предпочтениям конкретного человека. Вторая проблема – достижение общей прозрачности работы алгоритма, что с точки зрения пользователя выражается в непонимании предложенных рекомендаций. Немаловажной, третьей проблемой, является проблема безопасности и конфиденциальности, связанная с регламентами безопасности хранения данных с формированием ответственности за их защиту. Перечисленные этические проблемы, на наш взгляд, стоят особенно остро в поле разработки современных рекомендательных систем для фудтех индустрии.

Табл. 2. Сравнительный анализ алгоритмов машинного обучения для создания рекомендательных систем в фудтех индустрии (составлено автором).

Наименование	Достоинства	Недостатки
Коллаборативная фильтрация	<ul style="list-style-type: none"> – подходит для работы в условиях малого количества доступных данных; – не требует знания о продукте, только о предпочтениях пользователей; 	<ul style="list-style-type: none"> – не подходит для полностью новых продуктов, т.к. требуется наличие оценок
Content-based filtering	<ul style="list-style-type: none"> – подходит для рекомендаций новых продуктов; – объясняется причина конкретной рекомендации; 	<ul style="list-style-type: none"> – для реализации необходимо наличие подробной информации о продуктах;
Кластеризация	<ul style="list-style-type: none"> – эффективен для персонализации, позволяет выделить группы с определенными предпочтениями; 	<ul style="list-style-type: none"> – требует определения подхода к кластеризации и способов разделения пользователей и продуктов; – сложность реализации;
Матричная факторизация	<ul style="list-style-type: none"> – высокая точность предсказаний; – эффективность работы с большим количеством данных; 	<ul style="list-style-type: none"> – существует потребность в значительных вычислительных ресурсах; – сложность интерпретации результатов;
Reinforcement Learning	<ul style="list-style-type: none"> – гибкость и изменчивость рекомендаций; – учитывает наибольшее количество факторов; 	<ul style="list-style-type: none"> – требует больших объемов данных для обучения; – сложность реализации;
Глубокое обучение	<ul style="list-style-type: none"> – учитывает сложные зависимости; – формирует точные прогнозы; 	<ul style="list-style-type: none"> – существует потребность в значительных вычислительных ресурсах; – сложность интерпретации результатов;

Таким образом, результаты проведенного исследования показывают, что применение машинного обучения для персонализации рекомендаций в фудтех индустрии является перспективной задачей, реализация которой позволяет разработчику осуществлять эффективную разработку заданных систем. Применение предложенных алгоритмов отличается собственными перспективами и недостатками, а выбор конкретной системы исходит из ситуации, предпочтений разработчика и долгосрочных ориентиров бизнеса, в том числе финансовых ограничений и иных факторов влияния. Перспективы будущих исследований – осуществить раскрытие применения данных алгоритмов при разработке мобильных приложений в индустрии фудтеха, что найдет отражение в дальнейших научных публикациях автора.

Список литературы

1. Попова Е.П., Леоненко В.Н. Прогнозирование реакции пользователей в социальных сетях методами машинного обучения // Научно-технический вестник информационных технологий, механики и оптики. – 2020. – №1. – С. 118-124.
2. Сапрыкин Д.А., Подольский К.В., Панов А.В. Возможности применения машинного обучения в мобильных приложениях для персонализированной рекомендации контента // Международный журнал гуманитарных и естественных наук. – 2023. – №12-4(87). – С. 80-83.
3. Черняков А.Н., Дибиров М.Ш. О некоторых способах построения рекомендательных систем онлайн-маркетинга на основе алгоритмов машинного обучения // Инновации и инвестиции. – 2023. – №6. – С. 351-356.
4. Смоленчук Т.В. Метод коллаборативной фильтрации для рекомендательных сервисов // Вестник науки и образования. – 2019. – №22-1(76). – С. 18-21.
5. Инамова Д.Б., Чиркина М.А. Сбор информации для рекомендательных систем // Форум молодых ученых. – 2019. – №1-2(29). – С. 49-54.
6. Никулин В.Н., Прозорова Т.Г. Два алгоритма на основе техники стохастического градиентного спуска для рекомендательных систем // Вестник Пермского университета. Серия: Математика. Механика. Информатика. – 2014. – №3(26). – С. 48-56.
7. Харитонов Ю.С., Савина В.С., Паньини Ф. Предвзятость алгоритмов искусственного интеллекта: вопросы этики и права // Вестник Пермского университета. Юридические науки. – 2021. – №53. – С. 488-515.

References

1. Popova E.P., Leonenko V.N. Predicting user reactions in social networks using machine learning methods // Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics. 2020, no. 1, pp. 118-124.
2. Saprykin D.A., Podolsky K.V., Panov A.V. Possibilities of using machine learning in mobile applications for personalized content recommendation // International Journal of Humanities and Natural Sciences. 2023, no. 12-4(87), pp. 80-83.
3. Chernyakov A.N., Dibirov M.Sh. On some ways to build online marketing recommendation systems based on machine learning algorithms // Innovations and investments. 2023, no. 6, pp. 351-356.
4. Smolenchuk T.V. Collaborative filtering method for recommendation services // Bulletin of Science and Education. 2019, no. 22-1(76), pp. 18-21.
5. Inamova D.B., Chirkina M.A. Collection of information for recommendation systems // Forum of young scientists. 2019, no. 1-2(29), pp. 49-54.
6. Nikulin V.N., Prozorova T.G. Two algorithms based on the stochastic gradient descent technique for recommendation systems // Bulletin of Perm University. Series: Mathematics. Mechanics. Computer science. 2014, no. 3(26), pp. 48-56.
7. Kharitonova Yu.S., Savina V.S., Pagnini F. Bias of artificial intelligence algorithms: issues of ethics and law // Bulletin of Perm University. Legal sciences. 2021, no. 53, pp. 488-515.

Кукитз Паул Васильевич – технический директор	Kukitz Paul Vasilevich – CTO Restomania
e-mail: paulkukitz@gmail.com	

Received 29.05.2024