Моделирование и анализ данных
2020. Том 10. № 4. С. 17–30
DOI: https://doi.org/10.17759/mda.2020100402
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2020. Vol 10, no. 4, pp. 17–30
DOI: https://doi.org/10.17759/mda.2020100402
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

УДК 004.85

Прогнозирование покупки товара, показанного клиенту рекомендательной системой

Парфенов П.А.*
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0001-5995-347X
e-mail: pentalbymf@mail.ru

Тимофеева А.А.**
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0001-7043-3715
e-mail: alena195101@yandex.ru

Сологуб Г.Б.***
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

Алексейчук А.С.****
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация

Для цитаты:
Парфенов П.А., Тимофеева А.А., Сологуб Г.Б., Алексейчук А.С. Прогнозирование покупки товара, показанного клиенту рекомендательной системой // Моделирование и анализ данных. 2020. Том 10. № 4. С. 17–30. DOI: https://doi.org/10.17759/mda.2020100402

***Парфенов Павел Андреевич,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0001-5995-347X, e-mail: pentalbymf@mail.ru

****Тимофеева Алена А.,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация ORCID: https://orcid.org/0000-0001-7043-3715, e-mail: alena195101@yandex.ru

*****Сологуб Глеб Борисович,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail: glebsologub@ya.ru

******Алексейчук Андрей Сергеевич,** Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация

В работе рассматриваются различные методы улучшения рекомендательных систем. Проводится сравнительный анализ двух моделей для решения задач классификации: случайного леса (Random Forest) и CatBoostClassifier. Исследование выполнялось на данных истории покупок клиентов компании Ozon. Были использованы стандартные методы, часто применяемые в рекомендательных системах. Были реализованы методы коллаборативной фильтрации, косинусная схожесть товаров от просмотров клиента за одно посещение сайта, схожесть текстовых данных. Для оценки результатов использовались специальные метрики, оценивающие качество предсказаний первых k объектов из рекомендаций: Mean average precision (map@K) и Recall at K (recall@k). При генерации дополнительных признаков, основанных на различных методах, выявляющих схожесть объектов, отмечается увеличение качества прогнозов моделей. Модель CatBoostClassifier показала наилучшие результаты.

Ключевые слова: рекомендательные системы, машинное обучение, бинарная классификация, методы коллаборативной фильтрации, косинусная схожесть, map@K, recall@k.

1. ВВЕДЕНИЕ

В данной работе рассматриваются методы улучшения рекомендательных систем на примере компании Ozon, которая представляет собой один из ведущих интернет-магазинов на российском рынке. Для подобного интернет-магазина очень важно помогать покупателям находить необходимые товары быстрее, предоставляя возможность клиентам затрачивать наименьшее количество усилий и времени на поиски нужной вещи. С этой задачей отлично справляется рекомендательная система. Рекомендательная система – это система, определяющая рекомендации по предметам, которые могут быть полезны пользователю [1]. Так, когда человек смотрит или ищет что-то на сервисе, система выявляет то, что может ему понравиться и предлагает это, благодаря чему пользователь может быстрее найти необходимое, у него останутся хорошие впечатления о совершенной покупке, а магазин увеличит продажи за счет сбыта большего количества товаров. Одним из способов улучшения взаимодействия между сервисом, предоставляющим услуги, и клиентом, желающим ими воспользоваться, может быть улучшение рекомендательной системы, чтобы лучше удовлетворять его потребности.

В современном мире любая компания, которая предоставляет свои услуги через сеть интернет, сталкивается с проблемой выдачи релевантного контента своим пользователям. Рекомендация является релевантной, если она соответствует запросам пользователя [2]. Рекомендательные системы являются неотъемлемой частью продукта, через который происходит взаимодействие с будущими покупателями. Узнавая профиль клиента, его предпочтения и интересы, можно находить и выдавать для него необходимые ему услуги и товары. Большинство современных сервисов, предоставляющих свои услуги через интернет, используют системы рекомендаций. Например, компания Amazon. Она является одной из крупнейших платформ электронной коммерции, которая предоставляет множество услуг, от интернет-магазина до онлайн кинотеатра.

Amazon понимает, насколько важно правильно взаимодействовать с пользователем, и еще в 2000 годах компания реализовала свою первую рекомендательную систему и до сих пор продолжает работу по улучшению своих рекомендаций. Компания верит, что рекомендации должны сделать опыт взаимодействия с сервисом лучше [3].

В некоторых отраслях рекомендательные системы особенно важны, так как при эффективной настройке рекомендаций они могут помочь развитию бизнеса. Например, в онлайн-кинотеатре Netflix считают, что основой их бизнеса является именно рекомендательная система. Их рекомендательная система помогает компании в удержании клиентов: когда зритель начинает смотреть фильм, кинотеатр помогаем ему найти что-то интересное в течение нескольких секунд, предотвращая отказ от сервиса в пользу альтернативного варианта развлечения. Также благодаря рекомендациям определится подходящая категория пользователей для специфических фильмов, которые при попытке транслирования на традиционном телевидении принесли бы убытки, так как не нашли бы большого отклика среди зрителей и не смогли бы поддерживать значительный доход от рекламы [4]. Хорошо настроенная рекомендательная система позволяет удерживать старых пользователей и на основании их позитивного опыта привлекать новых. Поэтому развитие и улучшение рекомендаций очень важная задача, решение которой полезно как для бизнеса, так и для клиентов.

2. ПОСТАНОВКА ЗАДАЧИ

В данной работе рассматривается рекомендательная система компании Ozon. Когда покупатель заходит на сайт и страницу какого-нибудь товара, он видит описание товара, его цену и отзывы (рис.1). Внизу страницы товара есть графа «Рекомендуем также», в которой предлагаются похожие вещи, которые могут заинтересовать клиента (рис.2). Над этой системой рекомендаций и будет проводиться работа.

Рис. 1. Страница товара и его описание

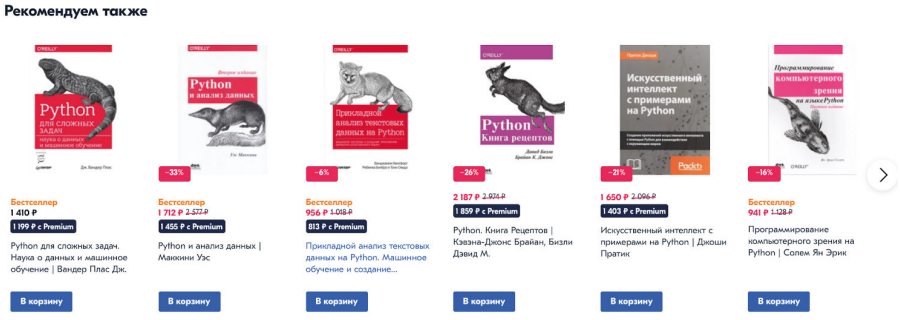

Рис. 2. Рекомендации к просмотренному товару

Система рекомендаций должна показывать, какие товары будут для покупателя наиболее актуальны. Насколько товар подходит пользователю, определяется данными о продажах товара и по истории покупок пользователя. Зная эту информацию, можно реализовать модель предсказания покупок, которые будут рекомендоваться клиенту. Полученный прогноз будет использоваться в качестве критерия для ранжирования, чтобы наиболее релевантные товары показывались первыми.

Определение рекомендаций товаров сводится к решению задачи ранжирования. Главным объектом является тройка элементов, на основании которого будет рассчитываться вероятность покупки:

$$X = \{c, i, r\}_{k=1}^{N},$$

где c – клиент, i – товар, который смотрит клиент, r – рекомендуемый товар.

Далее для каждого клиента (c) и товара (i) формируется свой список рекомендаций:

$$(c, i) \to r.$$

Для улучшения продаж рекомендуемых товаров необходимо, чтобы на первых местах в списке рекомендаций стояли те товары, которые имеют наибольшую вероятность покупки для данной пары клиента и товара, который смотрит клиент. Поэтому необходимо отсортировать данный список по уменьшению вероятности покупки товара:

$$r = \{r_1 \succeq r_2 \succeq r_3 \succeq \ldots \succeq r_n\},$$

где $1, \ldots, n$ – количество рекомендуемых товаров в списке.

На сайте уже реализована своя модель рекомендаций, она достаточно хорошо работает. Полное изменение данной системы не является целью данной работы. Основная задача – улучшение прогноза модели путем преобразования исходных данных и разработки дополнительных признаков, а также выбора более подходящего алгоритма модели. Прогноз строится для каждой пары клиента и просмотренного товара и предсказывается вероятность покупки каждого из товаров списка рекомендаций. В дальнейшем это поможет получить более точные рекомендации.

3. РЕШЕНИЕ ПОСТАВЛЕННОЙ ЗАДАЧИ

В качестве базовой модели использовалась модель бинарной классификации Случайный лес (Random Forest). Для обучения и тестирования использовались исторические данные клиентов. Учитывалось, какие товары смотрел покупатель, что добавлял к себе в корзину, что рекомендовалось, был ли куплен рекомендованный товар или нет, а также рассчитывались дополнительные признаки в виде описательных метрик для рекомендованного товара. Ниже представлены некоторые из них:

- количество просмотров/добавлений в корзину рекомендованного товара;
- количество просмотров/добавлений в корзину рекомендованного товара в первый и последний день наблюдаемого периода;
- конверсия добавления в корзину рекомендованного товара (отношение количества добавлений в корзину к количеству просмотров).

Используя данную модель, было получено базовое качество прогнозов. Ориентируясь на него, необходимо было преобразовать и дополнить исходный набор данных, чтобы улучшить качество предсказаний.

В ходе работы исходные данные были дополнены историей пользователей в течении одного посещения интернет-магазина и текстовыми описаниями товаров. Для этих данных были разработаны новые признаки и реализованы несколько стандартных подходов к построению рекомендаций. Их можно поделить на следующие типы:

1) коллаборативные методы фильтрации;
2) вычисление похожести покупателей по их сессиям;
3) вычисление схожести товаров по текстовому описанию;
4) выявление популярности и новизны товаров.

Вычисляя все перечисленные выше признаки, основной объект (клиент, просмотренный товар, рекомендуемый товар) рассматривается комплексно, с разных сторон. Это добавляет в данные дополнительную информацию, отражающую различные зависимости в объектах. Это может помочь модели легче выявлять закономерности покупки относительно текущего товара и рекомендуемого.

После построения всех необходимых признаков проводилось обучение модели и сравнение полученного качества с предсказаниями старого решения.

4. РЕАЛИЗАЦИЯ МЕТОДОВ КОЛЛАБОРАТИВНОЙ ФИЛЬТРАЦИИ

Одним из основных базовых подходов к реализации рекомендаций является метод коллаборативной фильтрации [5]. В нем можно выделить два метода:

1) основанный на сходстве пользователей (user-based);
2) основанный на сходстве элементов (item-based).

Оба метода позволяют находить похожих пользователей по их взаимодействию с объектами и на основании их оценок подбирать подходящие рекомендации для рассматриваемого пользователя, но выявляют эту похожесть немного по-разному. Так,

метод сходства пользователей основывается на схожести самих клиентов. Этот подход определяет, купит ли пользователь рекомендованный товар, если пользователи, похожие на него, покупали этот товар. Метод сходства элементов рассчитывает похожесть самих товаров. Пользователь купит рекомендованный товар, исходя из того, насколько этот товар похож на те товары, которые покупались этим клиентом раньше.

Алгоритм построения user-based подхода основывается на работе с несколькими матрицами. Первая и основная матрица – это матрица взаимодействий клиента и товара, где каждая строка – это идентификатор покупателя, а столбец – идентификатор товара. В ней отражается, какие товары покупал каждый пользователь (рис.3).

R =

clientid \ itemid	0	1	2	3	4
0	1	0	0	1	0
1	0	1	0	1	1
2	0	1	0	1	0
3	1	1	0	0	1

Рис. 3. Матрица взаимодействий клиентов и товаров

Чтобы подобрать товары, которые покупали похожие пользователи, необходимо найти схожесть всех покупателей. Для этого использовалась стандартная мера косинусной схожести [6] между двумя объектами-векторами:

$$\text{similarity} = \cos(\theta) = \frac{\vec{\tilde{n}_0} \cdot \vec{\tilde{n}_1}}{\left\|\vec{\tilde{n}_0}\right\| \left\|\vec{\tilde{n}_1}\right\|},$$

где $\vec{\tilde{n}_0} \cdot \vec{\tilde{n}_1}$ – скалярное произведение векторов клиентов с id, равными 0 и 1 соответственно, $\left\|\vec{\tilde{n}_0}\right\| \left\|\vec{\tilde{n}_1}\right\|$ – произведение евклидовых норм векторов $\vec{\tilde{n}_0}$ и $\vec{\tilde{n}_1}$. Эти векторы представлены в виде строк матрицы взаимодействий на рис.3.

При расчете похожести всех клиентов мы получаем матрицу вида (рис.4):

C =

clientid	0	1	2	3
0	1	0.4	0.5	0.4
1	0.4	1	0.8	0.33
2	0.5	0.8	1	0.4
3	0.2	0.33	0.4	1

Рис. 4. Матрица похожести клиентов

Чтобы выявить, какие товары могут понравиться рассматриваемому пользователю (например, клиенту с id = 0), необходимо взять вектор строки с id нужного

пользователя из матрицы похожестей клиентов (рис.4) и скалярно перемножить на векторы столбцов товаров из матрицы взаимодействий (рис.3). Скалярным произведением двух векторов называется число, равное произведению модулей векторов, умноженное на косинус угла между векторами [7]:

$$\vec{a} \cdot \vec{b} = \left|\vec{a}\right| \cdot \left|\vec{b}\right| \cos \widehat{\vec{a}\vec{b}}.$$

Так получаются оценки того, что товары понравятся пользователю на основании того, покупали ли эти товары похожие пользователи.

Основанный на сходстве элементов (item-based) подход реализуется аналогичным образом с одним лишь изменением, что вместо матрицы похожести клиентов рассчитывается матрица схожести товаров. Все остальные действия остаются без изменений.

5. НАХОЖДЕНИЕ ПОХОЖЕСТИ ПОКУПАТЕЛЕЙ ПО ИХ ПОВЕДЕНИЮ НА САЙТЕ

Еще одной характеристикой пользователя является его поведение во время посещения интернет-магазина. Пусть такое посещение называется сессией. В ней отражается, какие товары клиент смотрел или добавлял в корзину с максимальным перерывом между действиями в 30 минут. Логика расчета схожести товаров в зависимости от пользовательской сессии заключается в том, что если одни и те же товары часто будут смотреться в течение одного посещения, то скорее всего эти товары похожи и их можно вместе рекомендовать. Для нахождения похожести объектов также будем использовать меру косинусной схожести. Для этого составляется матрица взаимодействия товаров в сессиях (рис.5), где отражается, какие товары попадали в какие сессии.

itemid \ sessionid	0	1	2	3
0	1	0	1	0
1	1	0	0	0
2	0	1	1	0
3	1	1	1	1
4	0	0	1	0

Рис. 5. Матрица взаимодействия товаров и пользовательских сессий

Далее попарно для всех строк этой матрицы (рис.5) применяется формула меры косинусной схожести, получая при этом матрицу похожести всех товаров в разрезе пользовательских сессий.

6. РАСЧЕТ СХОЖЕСТИ ПО ТЕКСТОВОМУ ПРЕДСТАВЛЕНИЮ

В полученных данных было представлено текстовое описание товаров: название товара и его описание. Поэтому было принято решение рассчитать косинусную схожесть по текстовому описанию.

Для того чтобы рассчитать косинусную схожесть, необходимо провести предобработку данных. Для этого ко всем имеющимся описаниям товаров применим основы NLP (natural language processing) при работе с текстом:

- токенизация – разбиение текста на токены, в данном случае на слова;
- обработка текста с помощью регулярных выражений;
- лемматизация – приведение слов к их нормальной форме;
- удаление стоп-слов;
- расчет TF-IDF.

TF-IDF – статистическая мера, которая показывает важность слова для конкретного документа [14].

$$\mathrm{tf}\left(t,d\right)-\mathrm{idf}\left(t,d,D\right)=\mathrm{tf}\left(t,d\right)\cdot\log\left(\frac{D}{df_t}\right)$$

tf (t,d) – частота слова t в документе d;

df_t – количество документов, содержащих слово t;

D – общее количество документов;

idf (t,d,D) – инверсия частоты встречаемости слова t в документах D.

Обработанные документы были преобразованы в векторы с помощью векторизованной модели из библиотеки sklearn. Полученные векторы представляли собой разреженную матрицу, содержащую веса для каждого слова каждого документа, имеющего размер $D \cdot n$, где D – количество документов, а n – количество признаков (уникальных слов). Теперь эти веса из матрицы использовались в качестве признака для каждого документа, а сходство между документами вычислялось с использованием косинусного сходства.

7. ПОСТРОЕНИЕ МОДЕЛИ

После подготовки набора данных для обучения и тестирования необходимо начинать использовать модель, настраивать параметры и улучшать качество.

Для решения задачи ранжирования необходимо сначала решить задачу бинарной классификации, в которой необходимо будет предсказывать вероятность добавления рекомендованного товара в корзину. Получив вектор вероятностей, появится возможность отсортировать товары в рекомендациях по убыванию.

Обучение происходило с помощью библиотеки CatBoost. Это метод машинного обучения, основанный на градиентном бустинге (англ. *gradient boosting*). Бустинг – это подход построения композиций, в рамках которого: базовые алгоритмы строятся

последовательно, каждый последующий алгоритм строится таким образом, чтобы исправить ошибки уже построенной композиции. Композиция – объединение N алгоритмов $b_1(x),\ldots,b_n(x)$ в один. Идея композиции заключается в том, чтобы сначала обучить N базовых алгоритмов, а затем усреднить полученные ответы. Градиентный бустинг является одним из лучших способов направленного построения композиции [8]. Главным преимуществом данной библиотеки является то, что она одинаково хорошо работает как с числовыми признаками, так и с категориальными. Данная библиотека имеет хорошую документацию, обширный функционал и проста в использовании, так как не требует особой подготовки модели [9].

Подбор гиперпараметров модели осуществлялся с помощью метода GridSearchCV из библиотеки sklearn [10].

8. МЕТРИКИ

Mean average precision ($map@K$) – одна из наиболее часто используемых метрик качества ранжирования. В $p@K$ и $ap@K$ качество ранжирования оценивается для отдельно взятого объекта. Идея $map@K$ заключается в том, чтобы посчитать $ap@K$ для каждого объекта и усреднить [11]:

$$map@K = \frac{1}{N}\sum_{j=1}^{N} ap@K_j,$$

где N – количество объектов.

Precision at K ($p@K$) – точность на K элементах:

$$p@K = \frac{\sum_{k=1}^{K} r^{\text{true}}(\pi^{-1}(k))}{K} = \frac{\text{количество релевантых элементов}}{K}.$$

Под $\pi^{-1}(k)$ понимается элемент, который в результате перестановки π оказался на k-ой позиции, r^{true} – принимает значения 0 и 1, в зависимости от релевантности элемента.

Average precision at K ($ap@K$) – среднее только для релевантных товаров:

$$ap@K = \frac{1}{K}\sum_{k=1}^{K} r^{\text{true}}\left(\pi^{-1}(k)\right)\cdot p@k,$$

где k – количество только релевантных элементов.

Recall at K ($recall@k$) – доля релевантных элементов, найденных в топ-k рекомендациях [12]:

$$recall@k = \frac{\text{рекомендованные } k \text{ товаров, которые релевантны}}{\text{общее количество релевантных товаров}}.$$

AUC ROC (площадь под кривой ошибок) – доля пар объектов вида (объект класса 1, объект класса 0), который алгоритм верно упорядочил, т.е. первый объект идет в упорядоченном списке раньше [15].

9. АНАЛИЗ РЕЗУЛЬТАТОВ

Важным моментом в анализе обученной модели, является оценка важности признаков. Важность признаков нужна для понимания своего алгоритма, почему он именно так определяет ответ. С помощью важности признаков можно узнать, какие признаки лучше всего влияют на результат модели [13]:

$$\text{feature importance} = \sum_{\text{trees, leafs}_F} (v_1 - \text{avr})^2 \cdot c_1 + (v_2 - \text{avr})^2,$$

$$\text{avr} = \frac{v_1 \cdot c_1 + v_2 \cdot c_2}{c_1 + c_2},$$

где c_1, c_2 представляют собой общий вес объектов в левом и правом листьях. Вес равен количеству объектов в каждом листе, если веса не были заданы; v_1, v_2 представляют собой значения формулы в левом и правом листьях.

В ходе работы над данными, было разработано 22 признака. На следующем рисунке (рис.6) представлена важность этих признаков для построенной модели. На оси абсцисс расположены значения важности признаков, а на оси ординат представлены названия признаков. Можно заметить, что некоторые признаки несут больше информации для модели, чем другие. Например, признак same_items_on_session_view является лучшим признаком, а us_based_view не несет никакой значимости для модели. Эти признаки являются значением схожести рекомендованного товара с просмотренным в зависимости от поведения клиента за сессию и схожести, по основанной на сходстве пользователей (user-based), соответственно.

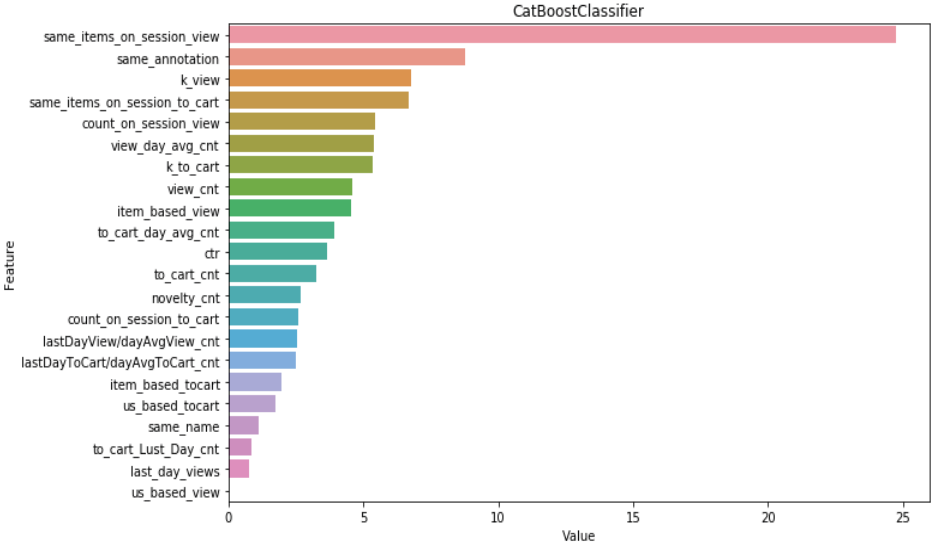

Рис. 6. Важность признаков

В качестве базовой модели был выбран алгоритм RandomForest, обученный на небольшом количестве признаков. На рисунке (рис.7) можно заметить, что обученная модель CatBoostClassifier сильно превзошла базовую модель.

Метрики	Baseline RandomForest	CatboostClassifier
AUC	0.55339	0.6345
Map@3	0.11874	0.1368
Recall@3	0.47102	0.5372

Рис. 7. Анализ результатов. Сравнение с базовой моделью на отложенной выборке

10. ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены возможные методы улучшения прогнозирования вероятности факта покупки рекомендованных товаров на примере интернет-магазина Ozon. В качестве исходных данных использовались данные об истории покупок клиентов, их поведения за сессию и текстовые описания товаров. Для построения прогнозов покупки рекомендованных товаров были использованы модели Random Forest и CatBoostClassifier. Для улучшения прогнозов были разработаны дополнительные признаки для обучения, а для сравнительного анализа моделей были реализованы специальные метрики, которые часто используются для оценки качества рекомендаций. В рамках задачи наилучшие результаты показал градиентный бустинг в реализации CatBoostClassifier.

Литература

1. *Francesco Ricci and Lior Rokach and Bracha Shapira*. Introduction to Recommender Systems Handbook // Springer Science+Business Media, LLC 2011. С. 1–10.
2. *Mizzaro Stefano*. Relevance: The Whole History // journal of the american society for information science, 1997. С. 810–820.
3. *Brent Smith and Greg Linden*. Two Decades of Recommender Systems at Amazon.com // the IEEE Computer Society, 2017. С. 10–17.
4. *Carlos A. Gomez-Uribe and Neil Hunt*. The Netflix Recommender System: Algorithms, Business Value, and Innovation // ACM Transactions on Management Information Systems, Vol. 6, No. 4, Article 13, 2015. С. 6–7.
5. *Е.Е. Пятикоп*. Исследование метода коллаборативной фильтрации на основе сходства элементов // Наукові праці ДонНТУ випуск 2 (18), Серія “Інформатика, кібернетика та обчислювальна техніка”, 2013. С. 109–110.
6. *Кристофер Д. Маннинг, Прабхакар Рагхаван, Хайнрих Шютце*. Введение в информационный поиск // Издательский дом “Вильямс”, 2011. С.138.
7. *Г.Г. Литова, Д.Ю. Ханукаева*. Основы векторной алгебры // Москва, 2009. С. 57.
8. *Jerome H. Friedman.* Greedy Function Approximation: A Gradient Boosting Machine // Technical Discussion: Foundations of TreeNet(tm), 1999. С. 39.
9. *CatBoost* [Электронный ресурс] // URL: https://neerc.ifmo.ru/wiki/index.php?title=CatBoost

10. *GridSearchCV* [Электронный ресурс] // Scikit-learn URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
11. *Gunnar Schröder*, *Maik Thiele, Wolfgang Lehner*. Setting Goals and Choosing Metrics for Recommender System Evaluations, 2011 С. 8.
12. *Ziwei Zhu, Jianling Wang, James Caverlee* // Improving Top-K Recommendation via Joint Collaborative Autoencoders, IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4 License, 2019. С. 7.
13. CatBoost Feature Importance [Электронный ресурс] // catboost URL: https://catboost.ai/docs/concepts/fstr.html#fstr
14. *Wen Zhang, Taketoshi Yoshida, Xijin Tang*. A comparative study of TFIDF, LSI and multi-words for text classification // Expert Systems with Applications, 2010. С. 8.
15. *Tom Fawcett*. An introduction to ROC analysis // Pattern Recognition Letters 27, 2006. С. 865.

Prediction the Probability of Purchases Recommended Items

Pavel A. Parfenov*
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0001-5995-347X
e-mail: pentalbymf@mail.ru

Alena A. Timofeeva**
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0001-7043-3715
e-mail: alena195101@yandex.ru

Gleb B. Sologub***
Moscow Aviation Institute (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000-0002-5657-4826
e-mail: glebsologub@ya.ru

Andrey S. Alekseychuk****
Moscow Aviation Institute (National Research University), Moscow, Russia

This paper discusses various methods for improving recommendation systems. A comparative analysis of two models for solving classification problems is performed: random forest and CatBoostClassifier. The research was performed on the data of the purchase history of Ozon customers. Standard methods that are often used in recommendation systems were used. We implemented collaborative filtering methods, cosine similarity of products from customer views per site visit, and similarity of text data. To evaluate the results, we used special metrics that evaluate the quality of predictions of the first k objects from the recommendations: Mean average precision (map@K) and Recall at K (recall@k). When generating additional features based on various methods that reveal the similarity of objects, an increase in the quality of model forecasts is noted. The CatBoostClassifier model showed the best results.

Keywords: recommendation systems, machine learning, binary classification, collaborative filtering methods, cosine similarity, map@K, recall@k.

For citation:
Parfenov P.A., Timofeeva A.A., Sologub G.B., Alekseychuk A.S.. Prediction the Probability of Purchases Recommended Items. *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2020. Vol. 10, no. 4, pp. 17–30. DOI: https://doi.org/10.17759/mda.2020100402 (In Russ., abstr. in Engl.).

***Pavel A. Parfenov,** Moscow Aviation Institute (National Research University), Moscow, Russian Federation, ORCID: https://orcid.org/0000-0001-5995-347X, e-mail: pentalbymf@mail.ru

****Alena A. Timofeeva.,** Moscow Aviation Institute (National Research University), Moscow, Russian Federation, ORCID: https://orcid.org/0000-0001-7043-3715, e-mail: alena195101@yandex.ru

*****Gleb B. Sologub,** Moscow Aviation Institute (National Research University), Moscow, Russian Federation, ORCID: https://orcid.org/0000-0002-5657-4826, e-mail: glebsologub@ya.ru

******Andrey S. Alekseychuk,** Moscow Aviation Institute (National Research University), Moscow, Russian Federation

References

1. Francesco Ricci and Lior Rokach and Bracha Shapira. Introduction to Recommender Systems Handbook// Springer Science+Business Media, LLC 2011, pp. 1–10.
2. Mizzaro Stefano. Relevance: The Whole History // journal of the american society for information science, 1997, pp. 810–820.
3. Brent Smith and Greg Linden. Two Decades of Recommender Systems at Amazon.com // the IEEE Computer Society, 2017, pp. 10–17.
4. Carlos A. Gomez-Uribe and Neil Hunt. The Netflix Recommender System: Algorithms, Business Value, and Innovation // ACM Transactions on Management Information Systems, Vol. 6, No. 4, Article 13, 2015, pp. 6–7.
5. E.E. Pyatikop. Study of the method of collaborative filtering based on the similarity of elements // Naukovi Pratsi DonNTU vipusk 2 (18), Series “Informatika, Kibernetika TA obchislyuvalna Tehnika”, 2013, pp. 109–110.
6. Christopher D. Manning, Prabhakar Raghavan, Heinrich schütze. Introduction to information retrieval // Publishing house “Williams”, 2011, pp. 138.
7. G.G. Litova, D.Y. Khanukaeva, Basics of vector algebra, Moscow, 2009, pp. 57.
8. Jerome H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine // Technical Discussion: Foundations of TreeNet(tm), 1999. P. 39.
9. CatBoost [Electronic resource] // URL: https://neerc.ifmo.ru/wiki/index.php?title=CatBoost
10. GridSearchCV [Electronic resource] // Scikit-learn URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
11. Gunnar Schröder, Maik Thiele, Wolfgang Lehner. Setting Goals and Choosing Metrics for Recommender System Evaluations, 2011 P. 8.
12. Ziwei Zhu, Jianling Wang, James Caverlee // Improving Top-K Recommendation via Joint Collaborative Autoencoders, IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4 License, 2019. P. 7.
13. CatBoost Feature Importance [Electronic resource] // catboost URL: https://catboost.ai/docs/concepts/fstr.html#fstr
14. Wen Zhang, Taketoshi Yoshida, Xijin Tang. A comparative study of TFIDF, LSI and multi-words for text classification // Expert Systems with Applications, 2010. P. 8.
15. Tom Fawcett. An introduction to ROC analysis // Pattern Recognition Letters 27, 2006. P. 865.