

Разработка рекомендательной системы на основе текущей сессии пользователя с использованием многоуровневого отбора кандидатов

Воробьева А.С., Мохов А.И.

Консультант: Кислинский В.Г.

Москва, 2020



Рекомендательная система — комплекс алгоритмов, программ и сервисов, задача которого предсказать, что может заинтересовать того или иного пользователя. В основе системы лежит информация о профиле человека и иные данные.

По сессии необходимо порекомендовать товары из интернет-магазина.

Пример сессий:

session	view	cart	order
0	id1, id2, id3	id2, id5, id6	id2, id7
1	id3, id6, id9	id11, id2, id6	id3, id1
2	id1, id2, id9	id11, id2, id6	id3, id1

Важный аспект задачи - известны лишь категории товаров, просмотры, добавления в корзину и заказы для конкретной сессии. **Нет привязки к пользователю!**

Примеры рекомендаций для просмотров

Защитное стекло Locase для Xiaomi Redmi Note 7, 2 штуки

★★★★☆ 343 отзыва Задать вопрос В избранное Сохранить Поделиться



Цвет: черный



Мышля на Озон показаний товаров!

Цвет рамки: Черный
Применение: На экран
Назначение: Для смартфонов
Страна-производитель: Китай
Цвет: Черный
Бренд: Locase
Рекомендовано для: Xiaomi
Тип: Защитное стекло
Вес в упаковке, г: 30
Перейти к описанию
Все товары Locase

В карточке товара находятся рекомендации, которые система автоматически сочла подходящими к текущему товару.

Рис.: Просмотренный товар

Рекомендуем также

Бестселлер
999₽
Защитное стекло Locase для Xiaomi Redmi Note 7, на...

В корзину

Бестселлер
2559 700P
Защитное стекло 5D Tempered Glass для Xiaomi...

В корзину

Бестселлер
3259 340P
Защитное стекло для Xiaomi Redmi Note 7

В корзину

Бестселлер
2499 1460P
Защитное стекло на Xiaomi Redmi Note 7 и Note 7 Pro /...

В корзину

Бестселлер
2999 1460P
Защитное стекло для Xiaomi Redmi Note 7 и Note 7 Pro

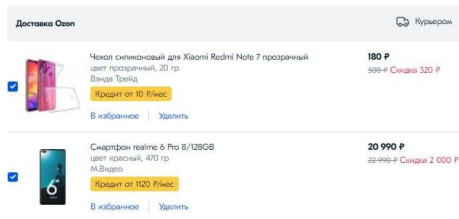
В корзину

Бестселлер
1859 260P
Защитное стекло Xiaomi Redmi Note 7 / на плоскую...

В корзину

Рис.: Рекомендации на основе просмотров

Примеры рекомендаций для товаров в корзине



После перехода в корзину система также формирует и показывает рекомендации, на основе товаров, которые лежат в корзине. Показывая релевантные для пользователя товары можно увеличить средний чек.

Рис.: Товары в корзине

Рекомендуем

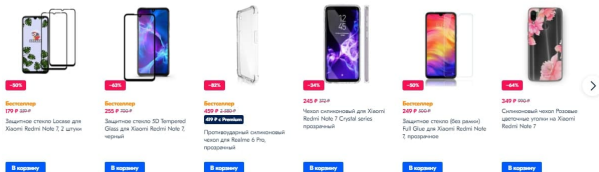


Рис.: Рекомендации на основе текущей корзины

Precision at K (P@K) - доля рекомендованных объектов, которые входят в K первых релевантных объектов.

Есть список рекомендация r длины N и задано $K \leq N$. Есть истинный список рекомендаций Y , и первые K его элементов обозначим Y_k .

Тогда можно посчитать P@k по формуле:

$$P@K = \frac{\sum_{i=1}^K \mathbb{1}_{Y_k}(r_i)}{K}$$
$$\mathbb{1}_{Y_k}(r_i) = \begin{cases} 1, r_i \in Y_k \\ 0, r_i \notin Y_k \end{cases}$$

Average precision at K (AP@K):

$$AP@K = \frac{\sum_{i=1}^K P@i}{K}$$

Целевая метрика - Average Precision @ K (AP@k):

$$Q(\hat{y}^i, y^i) = AP@K = \frac{1}{I} \sum_{i=1}^I AP@K^i \quad (1)$$

Усредняя данную метрику по всем сессиям, получаем Mean AP@k.

Введём отображение $f : id \rightarrow X$ - представление id в виде вектора в n -мерном пространстве.

Пусть матрицы X_{view}^i и X_{cart}^i - матричные представления сессии под номером i . Элементами матрицы являются вектора x_{view}^i и x_{cart}^i соответственно, которые представляют уникальные товары в данной сессии, получаемые из отображения f .

Паре (X_{view}^i, X_{cart}^i) соответствует вектор y^i , состоящий из id товаров, которые находятся в **заказе**.

Необходимо найти:

$$\hat{y}^i = A(X_{view}^i, X_{cart}^i) : Q(\hat{y}^i, y^i) \rightarrow \max \quad (2)$$

То есть найти такое преобразование A над исходными матрицами, что целевая метрика mAP@k достигала максимума.

В качестве решения предлагается использовать двухуровневый подход:

- 1 Отбор кандидатов с помощью простых моделей:
 - Обучение векторных представлений товаров (Word2Vec)
 - Матрица отношений
 - Использование популярных товаров в качестве кандидатов
- 2 Ранжирование с использованием градиентного бустинга [1, 2].

В результате получаем список рекомендаций для сессии.

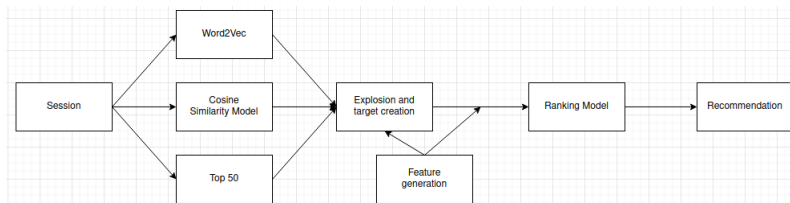


Рис.: Схема модели

Word2Vec - группа языковых моделей на нейронных сетях. Модели позволяют получить векторное представление (embedding) для слов (токенов)[3].

Пусть $w \in W$ - слово, в строковом формате. Введём отображение $f : W \rightarrow V$, переводящее множество W в V , где $V \in \mathbb{R}^n$.

Тогда в пространстве \mathbb{R}^n определена **метрика** d , задающая расстояние между любой парой $(x, y) (x \in \mathbb{R}^n, y \in \mathbb{R}^n)$

$$d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3)$$

Так же определено косинусное сходство между двумя элементами пространства \mathbb{R}^n :

$$\text{sim}(x, y) = \frac{(x, y)}{\|x\| \|y\|} = \frac{\sum_{i=1}^n x_i \cdot y_i}{\sqrt{\sum_{i=1}^n x_i^2} \cdot \sqrt{\sum_{i=1}^n y_i^2}} \quad (4)$$

Используя id товаров как токены, а сессии в качестве контекста можно получать векторные представления для товаров - **Prod2Vec**.

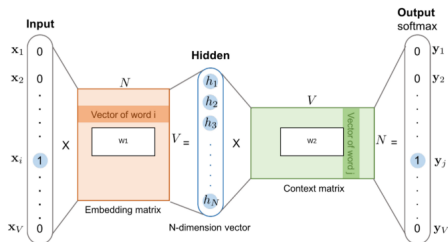
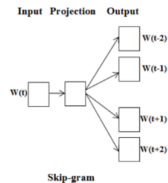
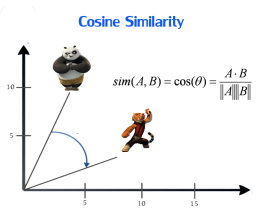


Рис.: Архитектура word2vec



Пусть I - множество товаров и $|I| = n$.

Множество сессий $S: S_j = \{j_1, j_2, \dots, j_m\} \subset I$.

Пример: две сессии $S_i = [1, 2, 3]$, $S_j = [3, 4, 5]$. Число уникальных элементов 5.

Матрица имеет размерность 5×5

	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	0	0
3	1	1	0	1	1
4	0	0	1	0	1
5	0	0	1	1	0

Выберем товары 2 и 3, посчитаем сходство между ними.

$$\text{sim}(c_2, c_3) = \frac{(c_2, c_3)}{\|c_2\| \|c_3\|} = \frac{1}{2\sqrt{2}}. \quad (5)$$

Таким образом составляется и общая матрица отношений.

Используем word2vec и матрицу схожестей для получения кандидатов (*первичные рекомендации*), а так же добавляем к ним список 50 самых популярных товаров из списка заказов.

После получения рекомендации для каждой сессии, таблица будет выглядеть следующим образом:

view	cart	order	prediction
$[id_{v1^1}, \dots, id_{vi^1}]$	$[id_{c1^1}, \dots, id_{cj^1}]$	$[id_{o1^1}, \dots, id_{ok^1}]$	$[p_1^1, \dots, p_n^1]$
...
$[id_{v1^m}, \dots, id_{vi^m}]$	$[id_{c1^m}, \dots, id_{cj^m}]$	$[id_{o1^m}, \dots, id_{ok^m}]$	$[p_1^m, \dots, p_n^m]$

где p - множество состоящее из id товаров.

Здесь уже можно посчитать $map@k$ - его значение колеблется от **0.01** до **0.015**.

mark зависит от порядка, поэтому необходимо учитывать релевантность товара из списка рекомендаций. Предлагается изменить таблицу следующим образом (операция Explode)

<i>session_id</i>	<i>order</i>	<i>prediction</i>	<i>target</i>
<i>session</i> ₁	<i>order</i> ₁	<i>p</i> ₁ ¹	<i>y</i> ₁ ¹
<i>session</i> ₁	<i>order</i> ₁	<i>p</i> ₂ ¹	<i>y</i> ₂ ¹
...
<i>session</i> ₁	<i>order</i> ₁	<i>p</i> _{<i>n</i>} ¹	<i>y</i> _{<i>n</i>} ¹
...
<i>session</i> _{<i>m</i>}	<i>order</i> _{<i>m</i>}	<i>p</i> ₁ ^{<i>m</i>}	<i>y</i> ₁ ^{<i>m</i>}
...
<i>session</i> _{<i>m</i>}	<i>order</i> _{<i>m</i>}	<i>p</i> _{<i>n</i>} ^{<i>m</i>}	<i>y</i> _{<i>n</i>} ^{<i>m</i>}

$$y_i^j = \begin{cases} 1, p_i^j \in order^j \\ 0, p_i^j \notin order^j \end{cases} \quad (6)$$

Если предлагаемый id находится в заказе для данной сессии, то ставим 1, если нет - 0.

Таким образом учитываем релевантность отдельного товара из рекомендации.

Метрика NDCG для сессии q , набора рекомендаций m_q и меток l_q определяется следующим образом:

$$NDCG(q_i) = \frac{DCG(q_i)}{IDCG(q_i)} \quad (7)$$

$$DCG(q_i) = \sum_{k=1}^{m_q} \frac{l_{qk}}{\log_2(k+1)} \quad (8)$$

$IDCG(q_i)$ - это значение для правильного ранжирования.

К сожалению, NCDG нельзя оптимизировать, т.к. она недифференцируема, поэтому вводится гладкая аппроксимация данной метрики **YetiRank**:

$$\mathbb{L} = - \sum_{i,j} \omega_{ij} \log \frac{e^{x_i}}{e^{x_i} + e^{x_j}} \quad (9)$$

Здесь (i, j) - индексы всех пар рекомендаций для сессии, ω_{ij} - вес пары рекомендаций, а x_i, x_j - предсказанные значения релевантности рекомендаций i и j соответственно.

Рассмотрим пример, где для двух сессии было получено 4 *первичных* предсказания:

<i>session_id</i>	order	prediction	relevance score
1	[2, 3, 1, 4]	4	0.1
1	[2, 3, 1, 4]	5	0.0
1	[2, 3, 1, 4]	1	0.3
1	[2, 3, 1, 4]	2	0.4
2	[3, 6, 10, 11]	10	0.05
2	[3, 6, 10, 11]	1	0.1
2	[3, 6, 10, 11]	6	0.5
2	[3, 6, 10, 11]	11	0.01

- ❶ группировка по id сессии
- ❷ сортировка с учётом предсказанных значений релевантности
- ❸ PROFIT

<i>session_id</i>	order	prediction
1	[2, 3, 1, 4]	[2, 1, 4, 5]
2	[3, 6, 10, 11]	[6, 1, 10, 11]

Благодаря переходу к задаче ранжирования удалось в разы улучшить целевую метрику $mAP@k$.

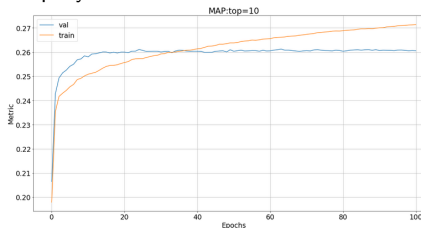


Рис.: $mAP@10$

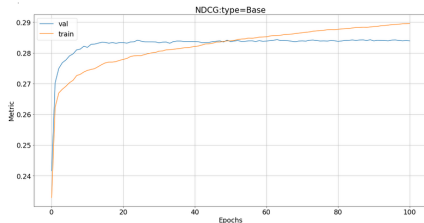


Рис.: NDCG

$$P(A|B) = \frac{P(AB)}{P(B)} \sim \frac{(A, B)}{\|B\|} \quad (10)$$

"Вероятность" покупки при условии что товары в сессии были добавлены в корзину или просмотрены.

$$P(Prediction|view) \sim \sum_{i=0}^k \frac{P(Prediction|view_i)}{P(view_i)}$$

$$P(Prediction|cart) \sim \sum_{i=0}^k \frac{P(Prediction|cart_i)}{P(cart_i)}$$

Схожесть всех товаров из корзины или просмотров с предсказанным товаром

$$sim(Prediction, view) = \sum_{i=0}^k sim(Prediction, view_i)$$

$$sim(Prediction, cart) = \sum_{i=0}^k sim(Prediction, cart_i)$$

Признаки имеют высокую корреляцию с целевой метрикой.

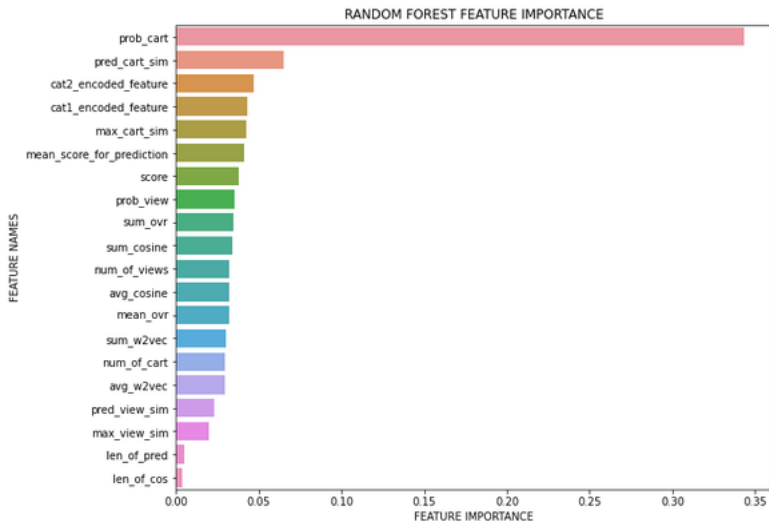


Рис.: Важность признаков в модели

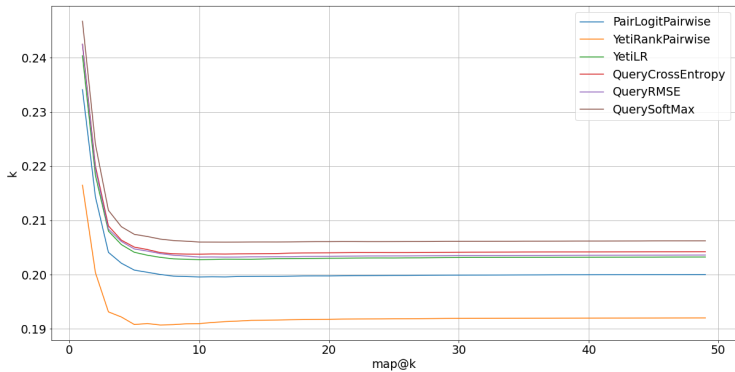


Рис.: Различные целевые функции для задач ранжирования

- [1] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin.
Catboost: gradient boosting with categorical features support.
arXiv preprint arXiv:1810.11363, 2018.
- [2] Andrey Gulin, Igor Kuralenok, and Dmitry Pavlov.
Winning the transfer learning track of yahoo!'s learning to rank challenge with yetirank.
In *Proceedings of the Learning to Rank Challenge*, pages 63–76, 2011.
- [3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean.
Distributed representations of words and phrases and their compositionality.
Advances in neural information processing systems, 26:3111–3119, 2013.