

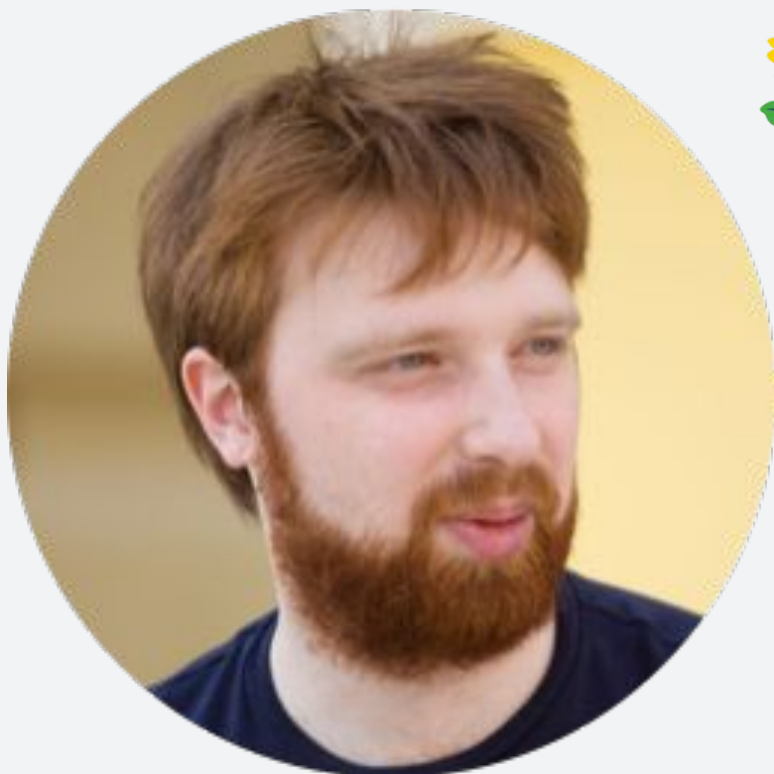
Skillbox

Введение в рекомендательные системы

Алексей Чернобровов

Консультант по Data Science

Алексей Чернобровов



ЛЕНТА



пульт



ру

skyeng

ВТБ

X5RETAILGROUP

@ mail.ru
group

Северсталь



Что такое рекомендательные системы?

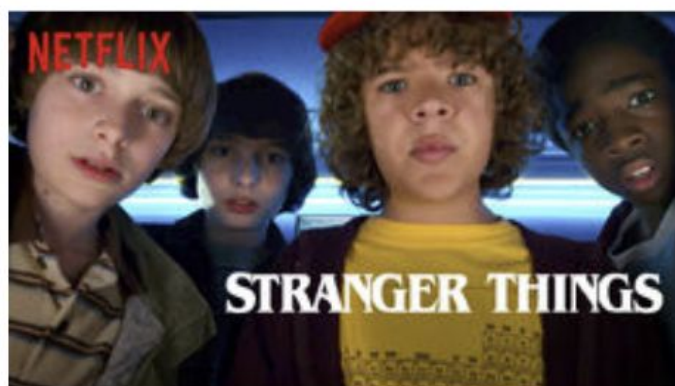
Рекомендательные системы

Рекомендательные системы — это программы, которые пытаются предсказать, какие объекты (фильмы, музыка, книги, новости, товары) будут интересны пользователю, на основе определенной информации о его профиле.



Рекомендательные системы

Netflix



Рекомендательные системы



Customers Who Bought This Item Also Bought



Apple iPad MC705LL/A
(16GB, Wi-Fi, Black) NEWEST
MODEL

★★★★★ (360)

\$509.95



3 Pack of Premium Crystal
Clear Screen Protectors for
Apple iPad

★★★★☆ (1,221)

\$1.69

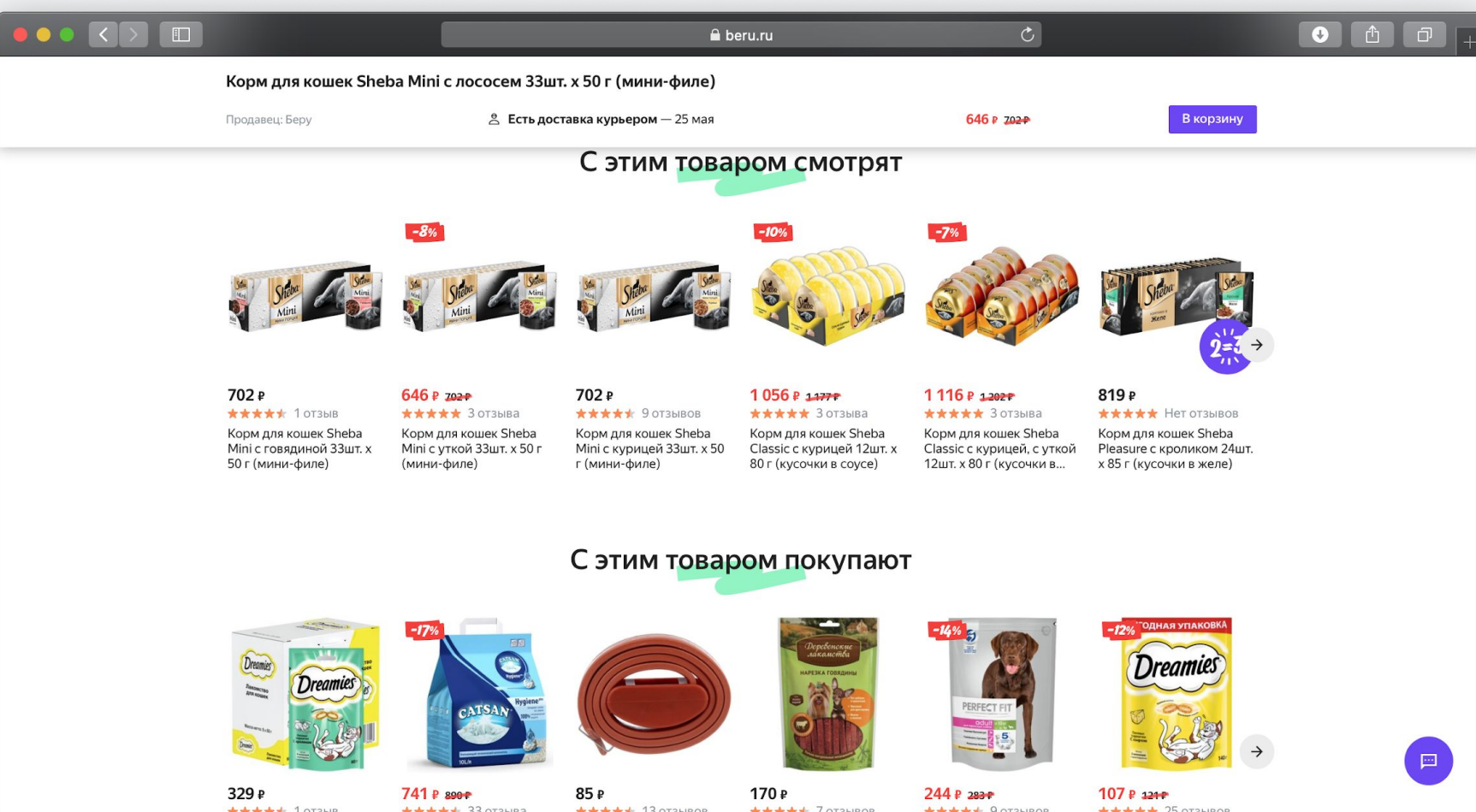


3 Pack of Universal Touch
Screen Stylus Pen (Red +
Black + Silver)

★★★★☆ (2,082)

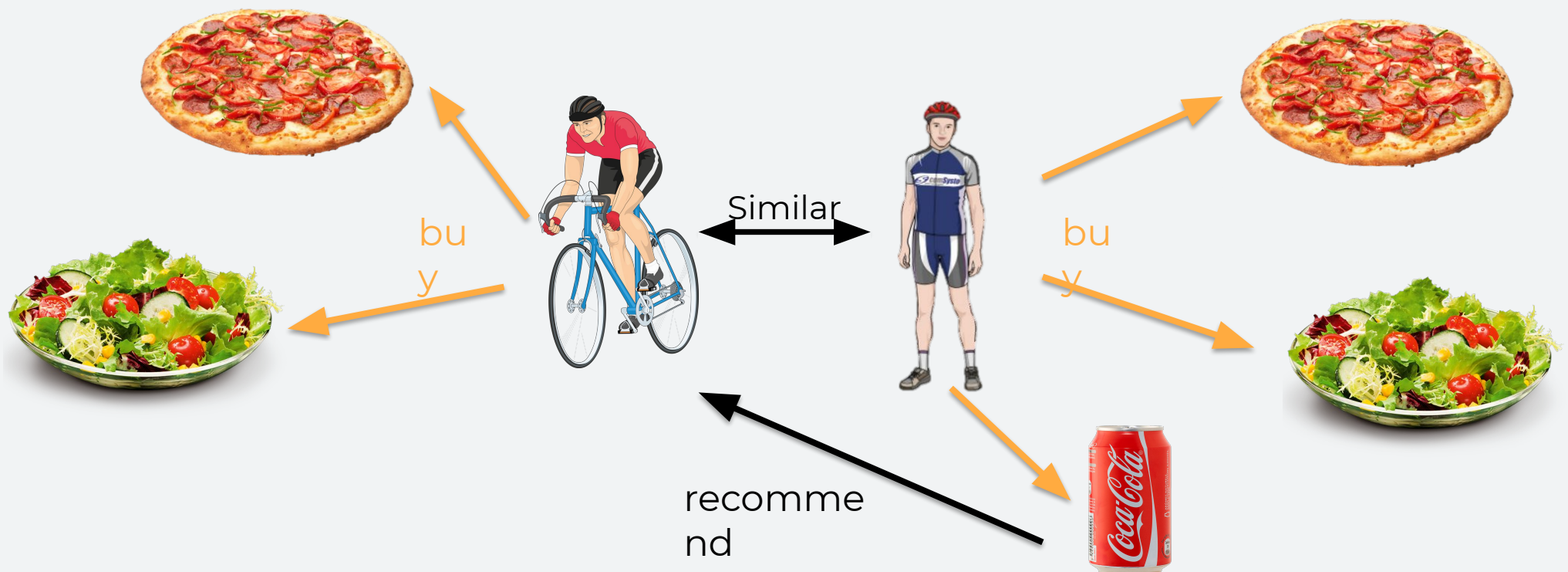
\$1.10





Рекомендательные системы

Такие системы значительно упрощают поиск релевантных продуктов и обогащают опыт пользователя. Множество компаний используют такие платформы для продвижения своих продуктов и услуг, руководясь запросами покупателей. В данном случае рекомендации основываются на историях поиска пользователей.



Рекомендательные системы

Релевантность — это мера того, насколько хорошо объект (документ, товар) удовлетворяет потребности пользователя в данный момент.

Например, пользователь, который вводит запрос в поисковую систему ожидает, что результаты будут соответствовать интену (поисковому намерению) и контексту (времени, месту, погодным условия) запроса.

Skillbox

Простейшие методы построения рекомендательных систем

Алексей Чернобровов

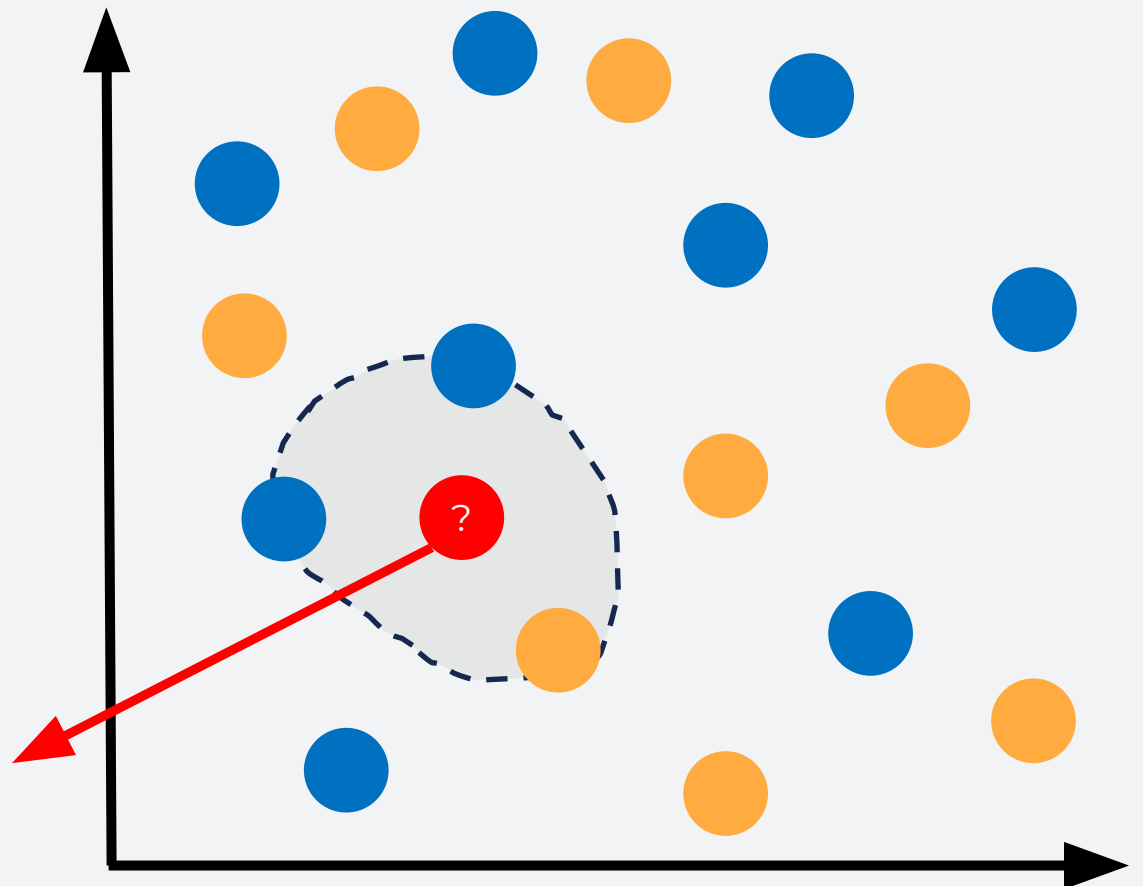
Консультант по Data Science

kNN (k Nearest Neighbor, или k ближайших соседей)

Метод ближайших соседей:

Давайте введём расстояние между пользователями и будем рекомендовать то, что нравится вашим соседям.

Предсказание на основе трех ближайших соседей



User-based kNN

	1+1	Три мушкетера	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

User-based kNN



	1+1	Три мушкетера	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

User-based kNN

	1+1	Три мушкетера	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	?
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

User-based kNN



Item-based kNN

	1+1	Три мушкетера	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	?
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

Item-based kNN

	1+1	Три мушкетера	12 стульев	Легенда №17
Алексей	10	9	1	7
Борис		9	2	?
Вова	1		6	
Коля	3		4	10
Петя		1		
Юля			3	6

Как оценить близость соседей?

Ключевым в алгоритме kNN является расстояние (близость). От того, как её задать, зависит результат.

Примеры расстояний:

1. Число совпавших оценок.
2. Корреляция Пирсона.
3. Косинусное расстояние.



Корреляция Пирсона

Корреляция Пирсона — классический коэффициент, который вполне применим и при сравнении векторов.

Основной его минус — когда пересечение по оценкам низкое, корреляция может быть высокой просто случайно.

$$\rho = \frac{\sum_i (\bar{x}_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Косинусное расстояние

Основная идея, на которой базируется расчет косинусного расстояния, заключается в том, что строку из символов можно преобразовать в числовой вектор. Если проделать эту процедуру с двумя сравниваемыми строками, то меру их сходства можно оценить через косинус между двумя числовыми векторами.

Из курса школьной математики известно, что если угол между векторами равен 0 (то есть векторы полностью совпадают), то косинус равен 1.

$$\text{similarity} = \cos(\theta) = \frac{XY}{\|X\| \|Y\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2}}$$

Skillbox

Практика на Surpriselib

Алексей Чернобровов

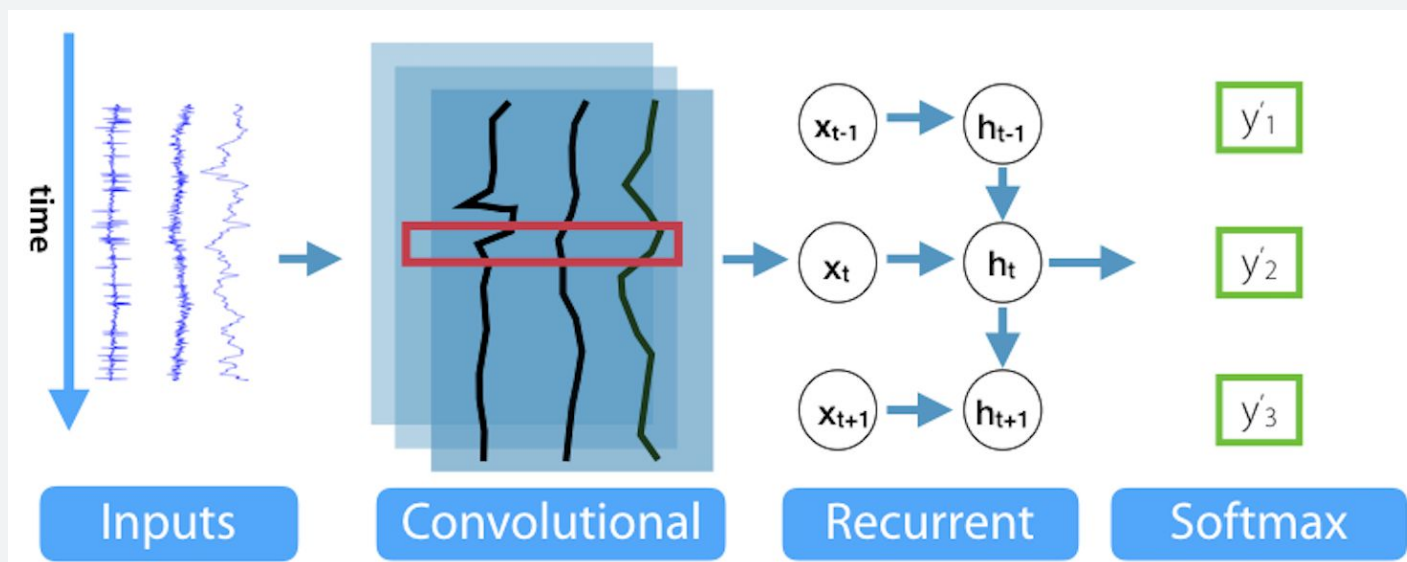
Консультант по Data Science

Обзор библиотек

- Turi Create,
- Implicit,
- Surpriselib.

Turi Create

Фреймворк для обучения моделей, основной идеей которого была простота в использовании и поддержка большого числа сценариев — классификация изображений, определение объектов, рекомендательные системы, и множество других.



github.com

turi.com

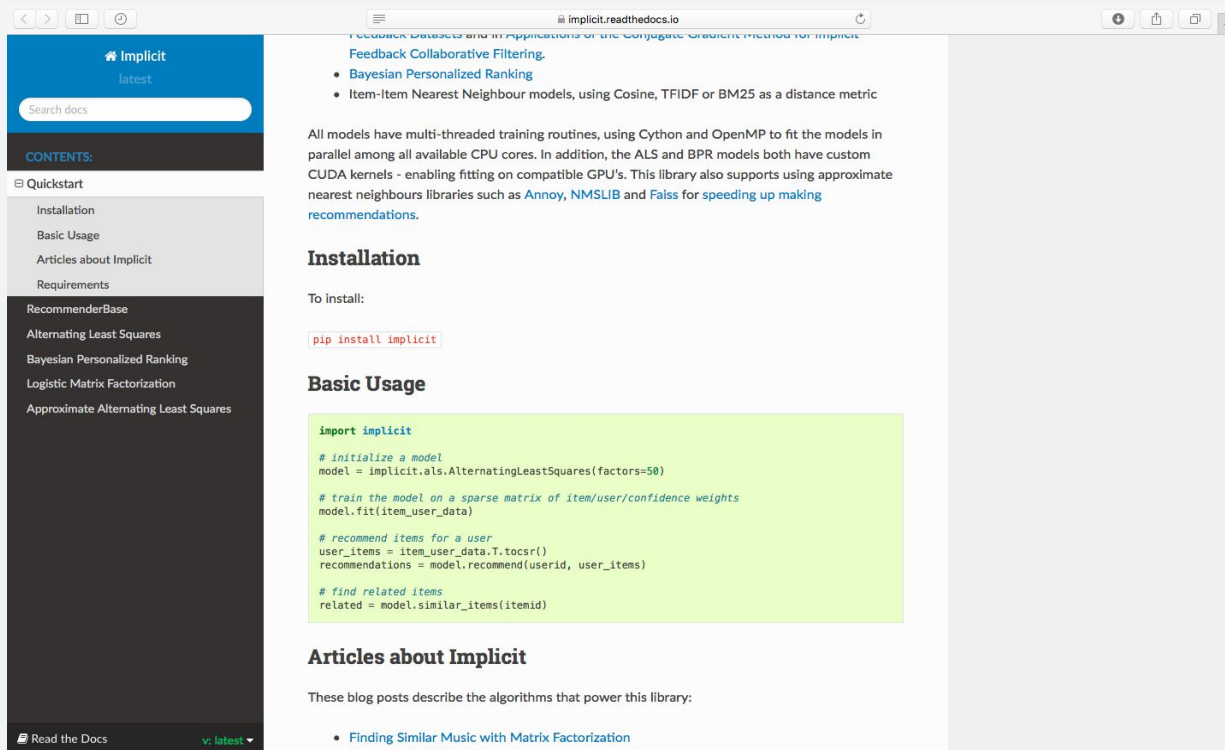


Only for Macintosh

Implicit

Библиотека на языке Python, в которой доступны несколько популярных алгоритмов рекомендаций:

- Alternating Least Squares (ALS),
- Bayesian Personalized Ranking и другие.

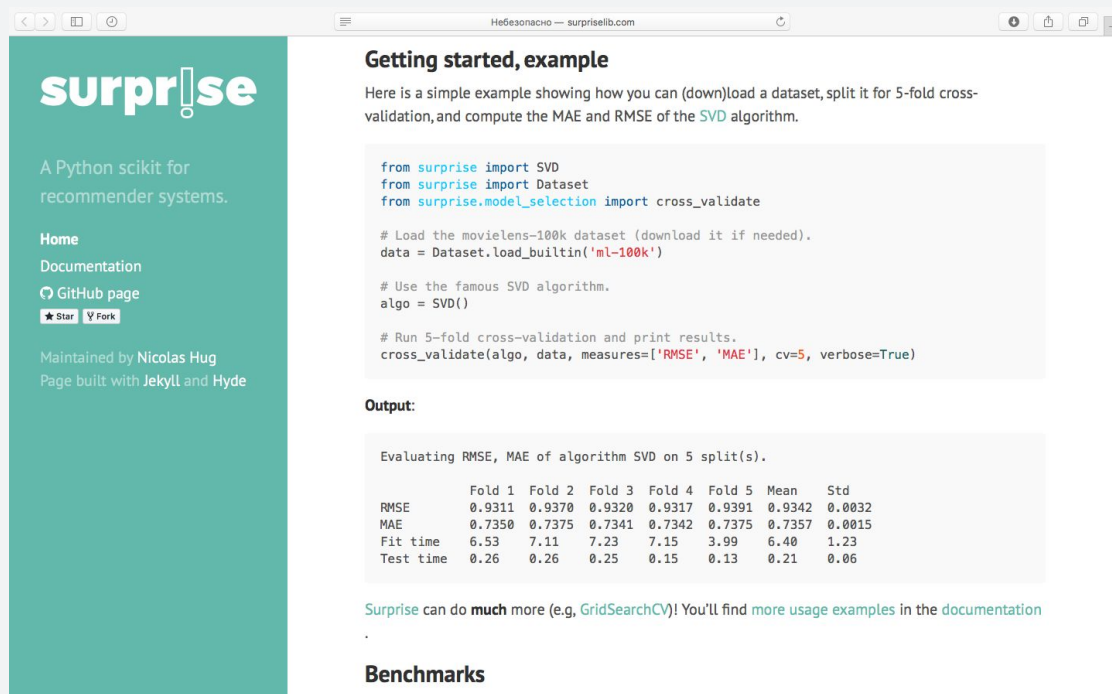


implicit.readthedocs.io

Surpriselib

Surprise — это пакет на scikit, создающий и анализирующий рекомендательные системы.

- Встроенные датасэты
- Предоставляет инструменты для оценки, анализа и сравнения производительности алгоритмов.
- Предоставляет различные готовые к использованию алгоритмы прогнозирования



The screenshot shows the website for surpriselib. On the left is a teal sidebar with the logo and navigation links. The main content area is titled 'Getting started, example' and contains a code block for loading the movielens-100k dataset and evaluating the SVD algorithm. Below the code is the output of the evaluation, showing RMSE, MAE, and fit/test times across 5 folds.

surprise

A Python scikit for recommender systems.

Home
Documentation
GitHub page
★ Star ▼ Fork

Maintained by Nicolas Hug
Page built with Jekyll and Hyde

Getting started, example

Here is a simple example showing how you can (down)load a dataset, split it for 5-fold cross-validation, and compute the MAE and RMSE of the [SVD](#) algorithm.

```
from surprise import SVD
from surprise import Dataset
from surprise.model_selection import cross_validate

# Load the movielens-100k dataset (download it if needed).
data = Dataset.load_builtin('ml-100k')

# Use the famous SVD algorithm.
algo = SVD()

# Run 5-fold cross-validation and print results.
cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```

Output:

```
Evaluating RMSE, MAE of algorithm SVD on 5 split(s).
```

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE	0.9311	0.9370	0.9320	0.9317	0.9391	0.9342	0.0032
MAE	0.7350	0.7375	0.7341	0.7342	0.7375	0.7357	0.0015
Fit time	6.53	7.11	7.23	7.15	3.99	6.40	1.23
Test time	0.26	0.26	0.25	0.15	0.13	0.21	0.06

Surprise can do **much** more (e.g. [GridSearchCV](#))! You'll find [more usage examples](#) in the [documentation](#).

Benchmarks

surpriselib.com

Skillbox

Разбор практики

Skillbox

Рекомендательные системы

Примеры рекомендательных систем

Рекомендации

- **Контента (фильмы, музыка, книги)**

Предложение нового контента, повышающего заинтересованность пользователей.



- **Товаров**

Предложение наиболее интересных товаров в интернет-магазине.



- **Событий (концертов, туров)**

Предложение наиболее интересных мероприятий для клиента.



Что мы знаем о пользователях?

Общая информация:

- устройство / браузер / размер экрана;
- регион;
- пол;
- дата рождения.

Поведенческие факторы (неявный отклик):

- просмотренные страницы (экраны);
- время на сайте или в приложении;
- клики;
- покупки.

Обратная связь (явный отклик):

- рейтинги;
- отзывы;
- «лайки».



Netflix Prize

Netflix Prize — это конкурс, в котором требовалось спрогнозировать оценку пользователями фильмотеки Netflix. Это была задача с явными рейтингами, оценки ставились по шкале от 1 до 5.

Были доступны следующие данные:

- Обучающие данные (training data set) содержат 100.480.507 оценок, которые 480.189 клиентов поставили 17.770 фильмам.
- Названия и годы выхода в прокат всех 17.770 фильмов.

Нужно было предсказать, какие оценки поставит пользователь тому или иному фильму.



Netflix Prize

Этот конкурс породил бум рекомендательных систем!

Определение победителя:

На скрытой части оказалось, что точность у этих команд совпадает до четвертого знака после запятой, поэтому победителя определила разница коммитов в 20 минут.

Победивший ансамбль использовал модели следующих классов:

- Регрессионная модель, основанная на средних оценках
- collaborative filtering — коллаборативная фильтрация
- Random Forests — предиктивная модель

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43

Метрики качества рекомендательных систем

Различные метрики

С исторического взгляда:

- **Метрики для регрессии:**
MAE, RMSE, MSE
- **Метрики поддержки принятия решения:**
Precision/Recall
- **Метрика, ориентированная на пользователя**
Охват, удержание пользователей, конверсии, клики



Уроки Netflix Prize

Netflix Prize — это конкурс, в котором требовалось спрогнозировать оценку пользователями фильмотеки Netflix.

Обучающие данные содержат 100 млн. оценок (от 1 до 5), которые 0,5 млн. клиентов поставили к 17 000 фильмам.

Точность прогноза оценивалась по **RMSE**.

Метрика:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (\hat{x}_i - x_i)^2}$$

x_i - истинное значение

\hat{x}_i - оценка

MAE

$$MAE = \frac{1}{m} \sum_{i=1}^m |\hat{x}_i - x_i|$$

RMSE

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (\hat{x}_i - x_i)^2}$$

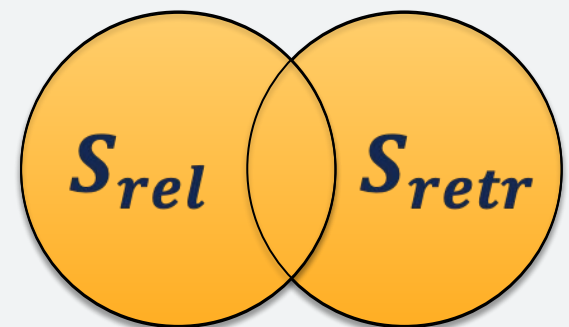
RMSE — метрика для предсказания оценки

Чем ниже значения **MAE** и **RMSE**, тем точнее механизм рекомендаций прогнозирует пользовательские рейтинги. Эти метрики удобны, когда рекомендации основаны на прогнозировании рейтинга или количестве транзакций. Они дают нам представление о том, насколько точны наши прогнозы и, в свою очередь, насколько точны наши рекомендации.

Precision

Precision — доля релевантных пользователю объектов относительно тех, которые ему показали.

$$precision = \frac{|S_{rel} \cap S_{retr}|}{|S_{retr}|}$$



S_{rel} — множество релевантных пользователю объектов.

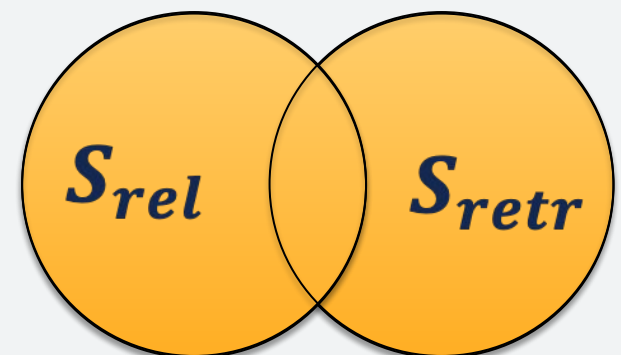
S_{retr} — множество показанных пользователю объектов.

Recall

Recall — доля релевантных объектов, показанных пользователю, относительно всех релевантных объектов.

Эту метрику можно интерпретировать, как вероятность того, что релевантный объект будет показан пользователю.

$$recall = \frac{|S_{rel} \cap S_{retr}|}{|S_{rel}|}$$



Бизнес-метрики

*Нужны ли бизнесу все
рассмотренные метрики?*

Бизнес-метрики

~~Нужны ли бизнесу все
рассмотренные метрики?~~

НЕТ!

Бизнес-метрики

Что нужно бизнесу?

- Конверсия
- Кликабельность
- Увеличение времени на сайте или в приложении
- LTV – ценность за период
- Стоимость привлечения клиента
- (CAC – customer acquisition cost)
- Коэффициент удержания клиента
- Время возмещения CAC (количество месяцев)
- Прибыль



Бизнес-метрики

Поэтому на практике чаще всего используются метрики, специально разработанные для каждой конкретной задачи в каждой компании.

Бизнес-метрики

Как правило, при разработке рекомендательных систем используются **прокси-метрики**.

Например, если в рекомендательном блоке можно показать всего 5 товаров, то хорошей метрикой может служить вероятность попадания товара в top-5.

А для реальной оценки бизнес-эффекта проводят **АБ-тесты** на пользователях.

Другие «метрики»

Также часто от рекомендательных систем ожидают других неформальных свойств.

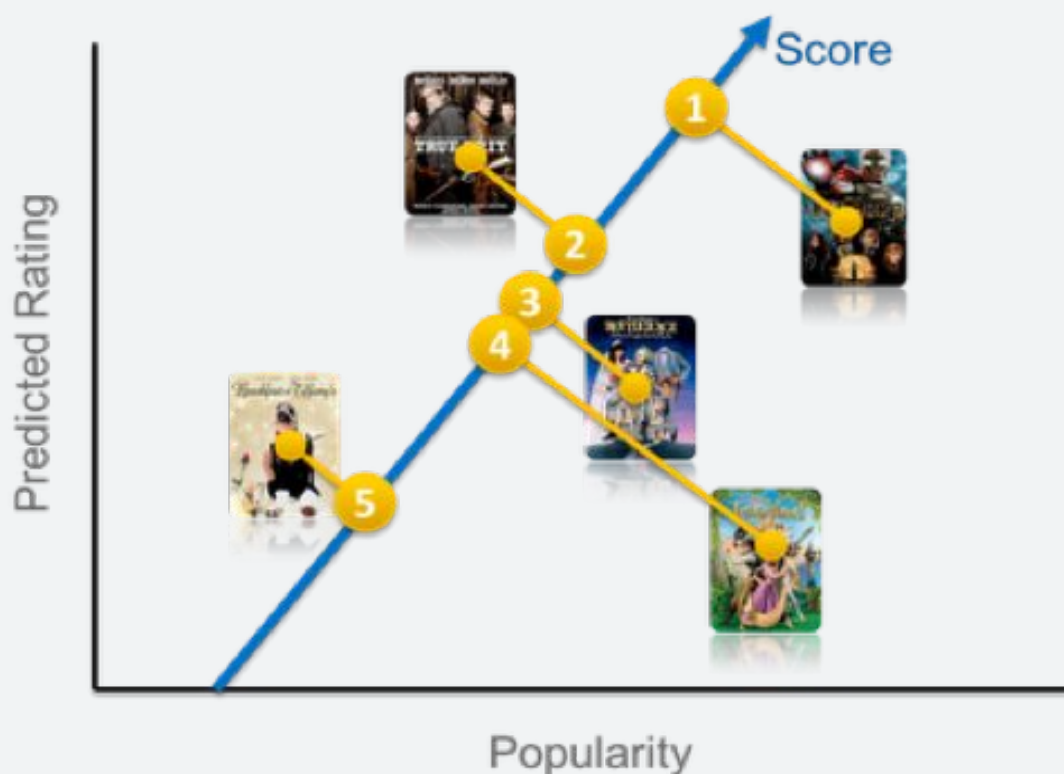
Например:

- Разнообразие
- Не тривиальных рекомендаций
- Покрытия запроса

Задача ранжирования

Задача ранжирования

Ранжирование — это класс задач машинного обучения с учителем, заключающихся в автоматическом подборе ранжирующей модели по обучающей выборке, состоящей из множества списков и заданных частичных порядков на элементах внутри каждого списка.



Задача ранжирования

Дано:

- Объекты: x_1, \dots, x_l
- Порядок на некоторых парах:

$$\{(i, j): x_i < x_j\}$$

Найти: Ранжирующую модель $a(x)$, такую что

$$x_i < x_j \implies a(x_i) < a(x_j)$$

Проще говоря, нужно построить модель, которая будет предсказывать правильный результат сравнения двух объектов.

И если сравнить все объекты между собой, то можно будет получить их ранги (порядковый номер).

И таким образом упорядочить их.

Примеры задач ранжирования

Примеры:

- Отсортировать (отранжировать) документы по релевантности
- Отсортировать письма по приоритету
- Отсортировать товары по вероятности покупки
- Предсказать места команд в чемпионате по футболу

Ранжирование в рекомендациях

- В задачах рекомендаций порядок устанавливается для пар (пользователь, объект)
- Порядок задан для каждого пользователя, определяется независимо

Задачи ранжирования

Как правило, порядок задается явной обратной связью:

- Для контента: оценками
- Для товаров: купил / не купил

Когда данных мало - используют неявную обратную связь:

- Для контента: время просмотра, частоту просмотра
- Для товаров: клики по товарам, добавления в корзину

Задачи ранжирования

Задачи ранжирования как самостоятельный класс задач.

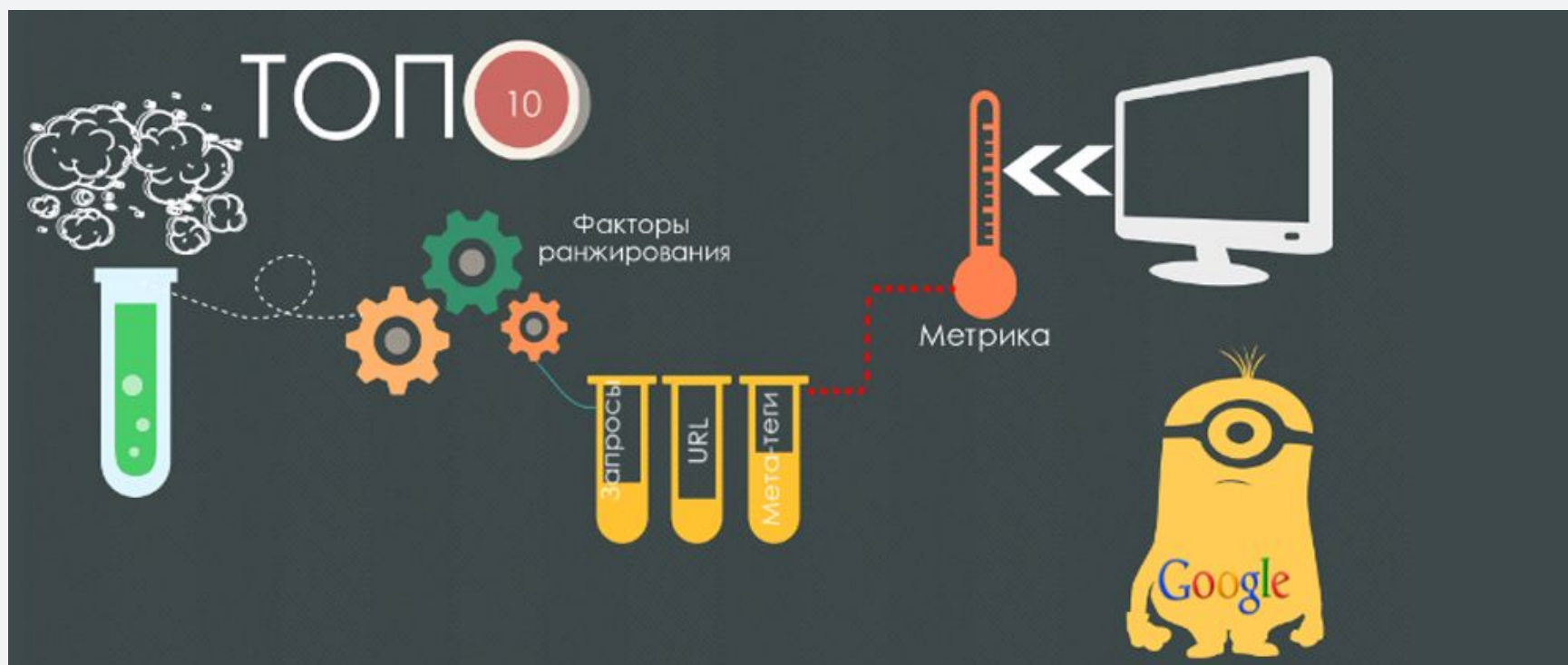
Методы для решения задач классификации и регрессии, в общем случае не подходят для решения задачи ранжирования.

Skillbox

Метрики ранжирования

Метрики ранжирования

1. Precision@n
2. Recall@n
3. MRR
4. NDCG@n
5. MAP



Recall@n

Recall@n – доля из первых n релевантных объектов, показанных пользователю относительно всех релевантных объектов.

Метрика не учитывает ни порядок, ни количество релевантных объектов.

AvgRecall@n – усреднение по всем пользователям.

$$\text{Recall@n} = \frac{|S_{rel}^n \cap S_{re}^n|}{|S_{rel}^n|}$$

$$\text{AvgRecall@n} = \frac{1}{Q} \sum_{q=1}^Q \text{Recall@n}(q)$$

Q — будем обозначать пользователей.

Precision@n

Precision@n — это доля релевантных пользователю объектов из первых n объектов.

$$Precision@n = \frac{|S_{rel}^n \cap S_{retr}^n|}{|S_{retr}^n|} = \frac{|S_{rel}^n \cap S_{retr}^n|}{n}$$

$$AvgPrecision@n = \frac{1}{n} \sum_{k=1}^n Precision@k \cdot rel(k)$$

AvgPrecision@n — показывает среднюю точность для первых n объектов.

Таким образом, эта метрика учитывает порядок документов.

Где k — позиция объекта в списке рекомендаций длины n.

$rel(k) = \{0,1\}$ — релевантность k-го объекта.

Бинарная функция, которая принимает значение 1, если объект релевантен и 0 — в противном случае.

Метрики ранжирования

MAP - Mean Average Precision

MAP помимо усреднения по n , дополнительно усредняет по всем пользователям или по всем запросам (q).

Это делается из соображения, что все пользователи или запросы равноценны.

MAP является достаточно популярной, учитывает и порядок, и количество релевантных объектов.

$$MAP@n = \frac{1}{Q} \sum_{q=1}^Q AvgPrecision@n(q)$$

Метрики ранжирования

MRR - Mean Reciprocal Rank

$$MRR = \frac{1}{Q} \sum_{q=1}^Q \frac{1}{rank_q}$$

$rank_q$ — означает
положение первого релевантного
объекта для пользователя q .

Метрики ранжирования

MRR - Mean Reciprocal Rank

Правильный порядок	Вариант прогноза	Ранг
А	В	3
Б	А	1
В	Б	2
Правильный порядок	Вариант прогноза	Ранг
А	Б	2
Б	А	1
В	В	3

$$MRR = \frac{1}{Q} \sum_{q=1}^n \frac{1}{rank_q}$$

Усредненная оценка
(1/3+1/2)/2 = 0.41

Идеальная оценка 1.

Метрики ранжирования

nDCG@n - Normalized Discounted Cumulative Gain

DCG@n является популярной метрикой в информационном поиске. Она учитывает и порядок, и количество релевантных объектов.

nDCG@n — это нормированная метрика.

nDCG@n = 1 означает, что объекты идеально отранжированны.

$$DCG@n = \sum_{k=1}^n \frac{rel(k)}{\log_2(k+1)}$$

$$nDCG@n = \frac{DCG@n}{IDCG@n}$$

$$IDCG@n = \sum_{k=1}^n \frac{1}{\log_2(k+1)}$$

Нормировочная константа

Что такое рекомендательные системы?

Методы ранжирования

- Pointwise (поточечный),
- Pairwise (попарный),
- Listwise (списочный).

Pointwise (поточечный)

Вместо предсказания порядка, будем предсказывать некоторую метрику, которая задается числом, и на которой сохраняется отношение порядка.

Например, для контента — это предсказание самой оценки или времени потраченного на контент.

Обратите внимание, что с точки зрения задачи ранжирования — нам нужно оценить только порядок, а не саму оценку.

Pairwise (попарный)

В этом случае задача обучения ранжированию аппроксимируется решением задач классификации. Каждое сравнение пары объектов рассматривается, как отдельная задача классификации.

Цель состоит в том, чтобы свести к минимуму среднее число инверсий в рейтинге.

Pairwise (попарный)

Минимизируем количество пар, на которых алгоритм совершает ошибку:

$$\sum_{x_i < x_j} [a(x_j) - a(x_i) < 0] \rightarrow \min$$

$L(M)$ – гладкая функция

$$\sum_{x_i < x_j} L(a(x_j) - a(x_i)) \rightarrow \min$$

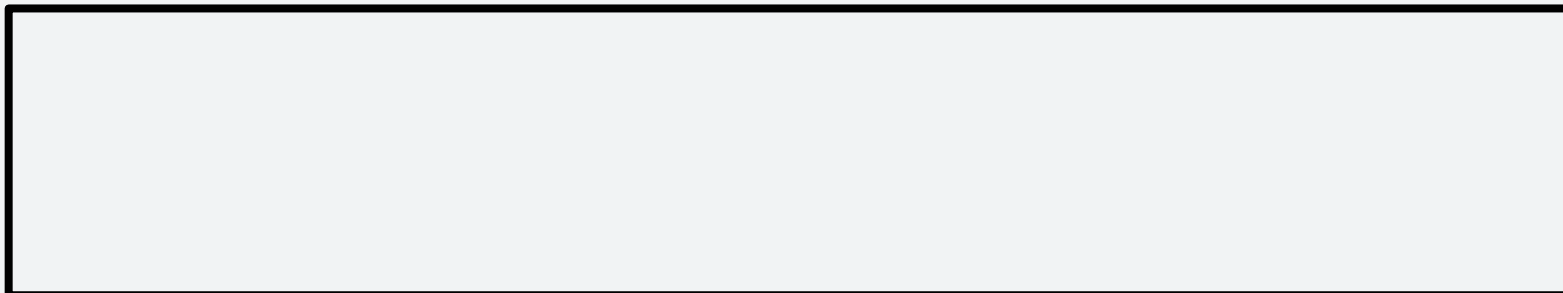
$L(M) = \log(1 + e^{-M})$ - метод RankNet

Listwise (списочный)

Списочный подход непосредственно пытается найти оптимальный порядок для всего списка объектов. Но в такой постановке задачи присутствует дискретный функционал, и с таким функционалом затруднительно работать напрямую, поскольку он не дифференцируется.

RankNet

Шаг стохастического градиентного спуска для линейной модели:



LambdaRank

Домножим стохастический градиент по паре (x_j, x_i) на $\Delta NDCG_{ij}$ (изменение **NDCG** при

Другие подходы

Существуют другие методы работы со списочным подходом, и на данный момент это еще открытая область машинного обучения, и почти каждый год выходят новые работы.

Их условно разделить на 2 типа:

1. Аппроксимация попарного подхода. Например, LambdaRank, SoftRank, AdaRank.
2. Использование рангов в явном виде. Иногда с использованием специфики конкретных задач. Например ListNet, ListMLE.

Выводы

Что мы рассмотрели:

- Рекомендательные системы, их типы и методы построения
- Обзор работы библиотек
- Метрики рекомендательных систем

**Спасибо за
внимание!**