

# Лекция 11

# Рекомендательные системы



**girafe**  
**ai**



# Персонализация важна в различных сервисах



+\$2.93 млрд дополнительно к бюджету компании после внедрения рекомендательной системы во все сервисы



80% того, что смотрят пользователи является результатом рекомендаций; итог: экономия \$1 млрд



Spotify достиг 96 миллионов премиальных подписок за 2018; средняя выручка с подписки – €4.89

# Что такое рекомендательная система?



## Примеры:

- Amazon
- Netflix
- Spotify
- Ozon

Множество областей применения: e-commerce, новости, туризм, развлечения, образование...

Цель: предсказать пользовательские предпочтения на основе его предыдущих действий

# В более общем смысле

Рекомендательные системы ставят целью восстановить **частично наблюдаемые взаимоотношения** между **двумя или более типами объектов**.

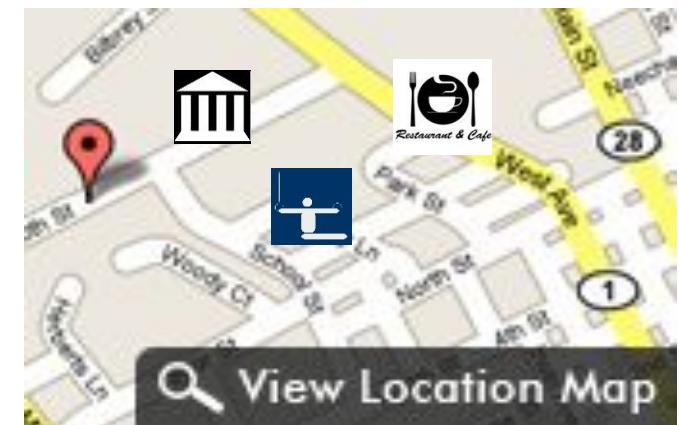
Социальные сети:  
пользователь ↔ пользователь



Данные на основе  
последовательностей:  
предмет → предмет  
(порядок имеет значение)

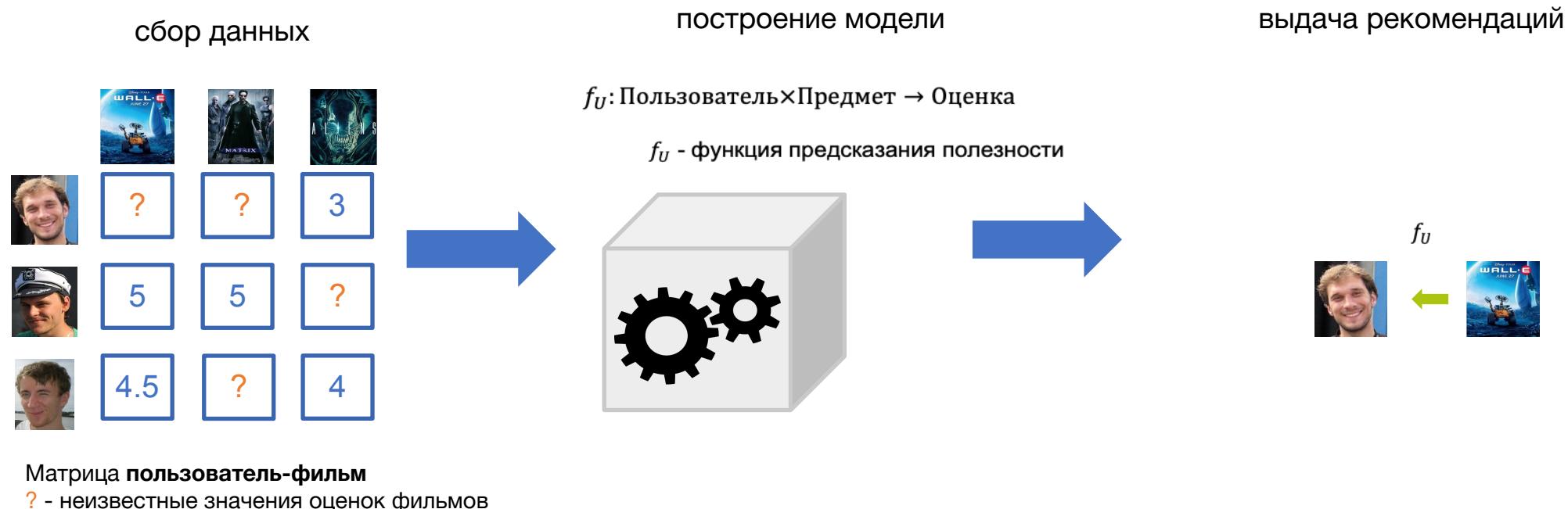


Множественные отношения:  
пользователь → действие → место



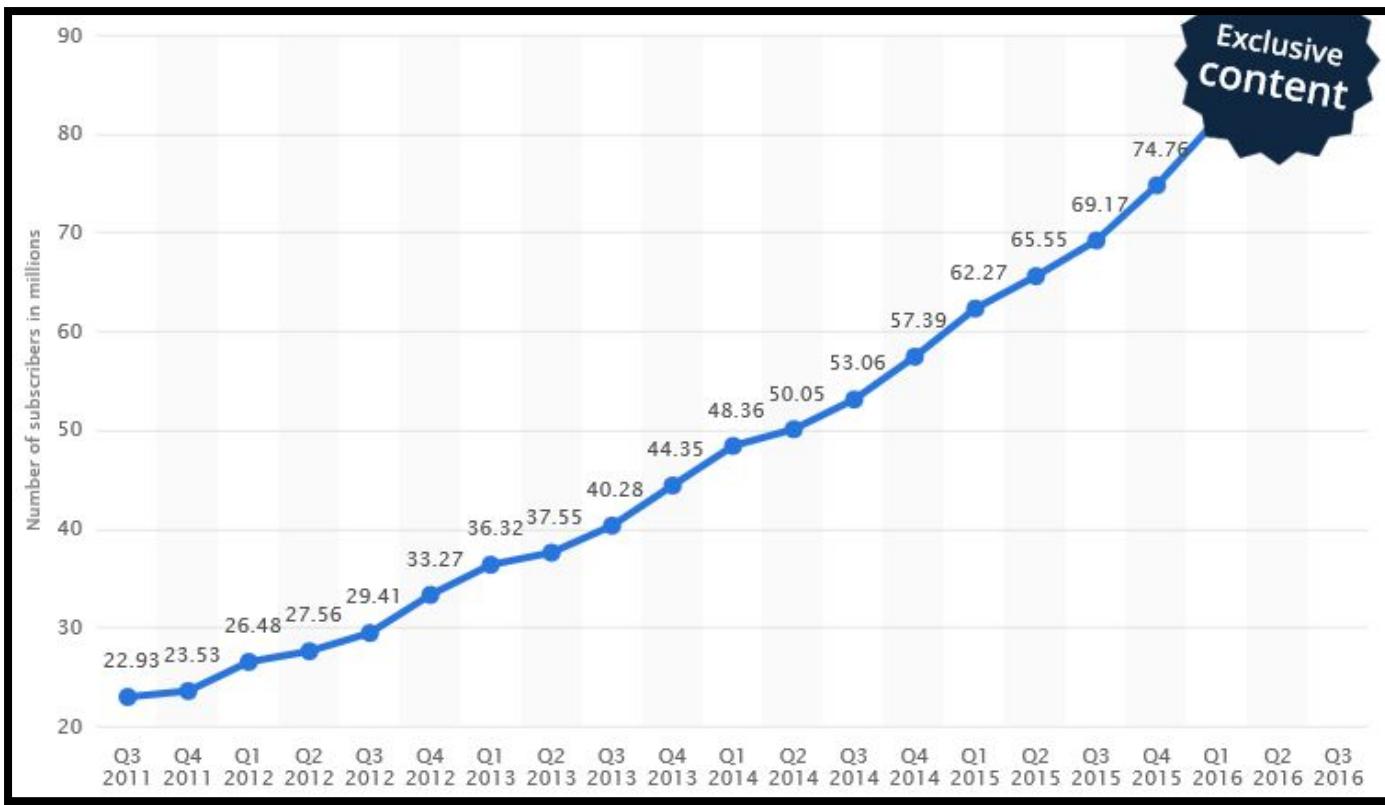
# Процесс создания и использования модели рекомендаций

**Цель:** предсказать пользовательские предпочтения на основе его предыдущих действий.

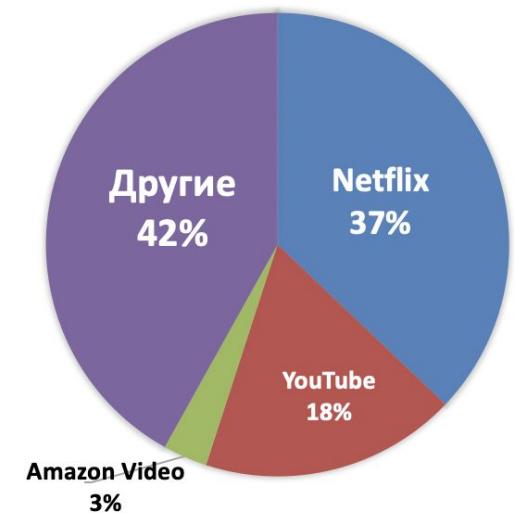


# Пример: развлечения

## Аудитория Netflix



## Распределение интернет трафика



## Примеры: финансовый сектор

Возможные сценарии:

- Индивидуальные рекомендации банковских продуктов
- Транзакции пользователей могут быть использованы для угадывания возможных интересов:
  - посетить торговый центр неподалеку
  - приобрести определенные продукты

Что это дает:

- увеличивает привязанность к банковским услугам
- позволяет удобным образом описать возможные интересы пользователей
- улучшает связность и надежность накопленной информации о пользователях
- дает новые инсайты для принятия бизнес-решений и формирования маркетинговой стратегии



Ведет к увеличению выручки в долгосрочной перспективе

# Смешные ошибки рекомендательных систем

“Если вам нравится книга Diversity Myth Питера Тиля,  
то вам может понравиться гиря”

this book is a compelling insider's tour of a world of speech codes, "dumbed-down" admissions standards and curricula, campus witch hunts, and anti-Western zealotry that masquerades as legitimate scholarly inquiry. Sacks and Thiel use numerous primary sources—the Stanford Daily, class readings, official university publications—to reveal a pattern of politicized classes, housing, budget priorities, and more. They trace the connections between such disparate trends as political correctness, the gender wars, Generation X nihilism, and culture wars, showing how these have played a role in shaping multiculturalism at institutions like Stanford. The authors convincingly show that multiculturalism is not about learning more; it is actually about learning less. They end their comprehensive study by detailing the changes necessary to reverse the tragic disintegration of American universities and restore true academic excellence.

[Read less](#)

[Report incorrect product information.](#)

Customers who viewed this item also viewed

AmazonBasics Enamel Kettlebell  
★★★★★ 187  
\$19.77 - \$67.99

“Ищете запчасти для двигателя? Вам могут пригодиться женские туфли ”

2.4 engine porsche

Categories

All eBay Motors Parts & Accessories Automotive Tools & Supplies Consumer Electronics

Condition

New (71) Used (92)

Price

\$ to \$

Format

All Listings (98) Auction (11) Buy It Now (74)

Item Location

Default Within 10 mi. 10 mi. or 13198 US Only North America Worldwide

Delivery Options

Free shipping

Find parts for your automobile

Select a vehicle: Select Make Any Model Any Year Any Trim Any Engine

\*\*1973 PORSCHE 911 2.4 L USED ENGINE ASSY. (911/91)\*\* \$4,495.00 Buy It Now 51 watching

Porsche 911 engine 2.4 liter 911S rebuilt 1972 long block (Fits: Porsche) WE have other 356 & 911 engines in stock / buy cores 2 \$9,900.00 or Best Offer 48 watching

Porsche 911 engine 2.4 liter real 1972 911E rebuilt \* Long Block\* rare motor WE have other 356 & 911 engines in stock / buy cores 2 \$9,900.00 or Best Offer

Merona Women's 3/4 to 2 Solid Leather Shoes

Merona Women's 3/4 to 2 Solid Shoes

Women's 3/4 to 2 Solid Heels

Related?

# Не такие смешные ошибки рекомендательных систем

## Циклическая обратная связь

Радикализация взглядов пользователей  
YouTube

## Bias

Алгоритмы могут смещены по переменным,  
которых нет в данных.



The New York Times @nytimes · Jun 8

Caleb Cain was a college dropout looking for direction. He was then pulled into YouTube's far-right universe, watching thousands of videos filled with conspiracy theories, misogyny and racism. "I was brainwashed."



The Making of a YouTube Radical

Caleb Cain was a college dropout looking for direction. He turned to YouTube, where he was pulled into a world filled with conspiracy theories...

nytimes.com

<https://habr.com/ru/post/437572/>

<https://www.nytimes.com/interactive/2019/06/08/technology/youtube-radical.html>

<https://www.fast.ai/2019/05/28/google-nyt-mohan/>

# Как оценить качество рекомендательных систем

# Поиск vs Рекомендации

Поисковая система использует запрос.

Рекомендательная система – поиск без запроса.

Google search results for "recommender system infrastructure".

Search bar: recommender system infrastructure

Filter bar: All, Images, News, Videos, Shopping, More, Settings, Tools

About 71,800,000 results (0.43 seconds)

Machine Learning for Recommender systems — Part 1 (algorithms ...  
https://medium.com/.../machine-learning-for-recommender-systems-part-1-algorithms... ▾  
Jun 3, 2018 - Recommender systems are one of the most successful and widespread application of machine learning technologies in business. There were ...

System Architectures for Personalization and Recommendation  
https://medium.com/.../system-architectures-for-personalization-and-recommendation-... ▾  
Mar 26, 2013 - A software architecture that handles large volumes of existing data, ... system diagram for recommendation systems in the following figure.

People also ask

- What is meant by recommender systems? ▾
- How does Amazon's recommendation system work? ▾
- What is content based filtering? ▾
- Is collaborative filtering machine learning? ▾

Feedback

Machine Learning for Recommender systems — Part 2 (Deep ...  
https://medium.com/.../machine-learning-for-recommender-systems-part-2-deep-reco... ▾  
Jun 7, 2018 - There are several ways how to utilize deep learning in recommender systems. Neural networks can be trained to predict ratings or interactions ...

# История Netflix prize: октябрь 2006 – июнь 2009



Условия соревнования: На основе предоставленной компанией информации об оценках пользователей выбранным фильмам, улучшить точность предсказания оценок хотя бы на 10% относительно внутреннего алгоритма Netflix

Приз: \$1,000,000

Соревнование продемонстрировало важность области для индустрии



## Проблемы:

- Умение точно предсказывать оценку ≠ умение выдавать релевантные рекомендации

Интересный факт:  
решение победителей не было  
использовано в produciton

# Оценки качества рекомендательных систем



Netflix использовал корень из среднеквадратичной оценки RMSE на оценках.

Проблемы RMSE:

- У ошибок в бизнесе разный вес: сравните 5 vs 3 и 3 vs 1. Так как фильм с оценкой 1 или 3 мы рекомендовать не будем, а с оценкой 5 будем.
- Не отражает бизнес суть задачи!

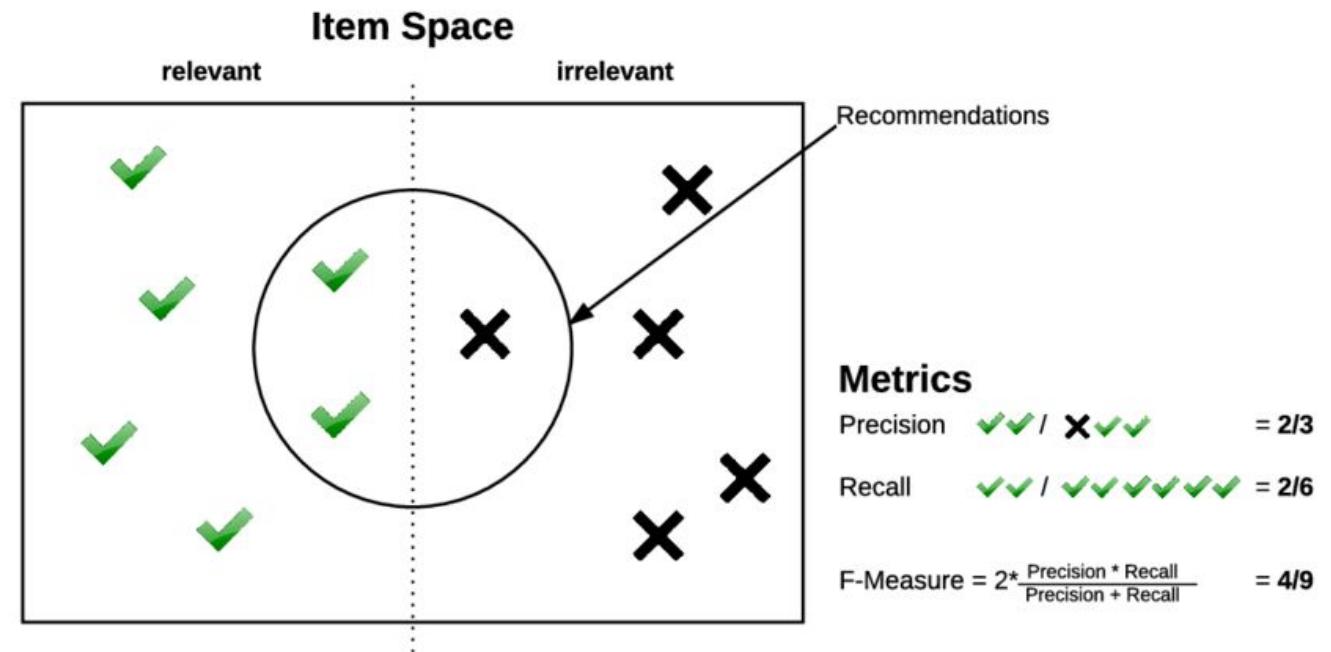
$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$$

# Точность и специфичность

**Precision, точность** – доля релевантных для пользователя объектов относительно всех, которые были ему показаны.

Мотивация: пользователю нельзя показать все наши продукты, блок предложений конечен.

**Recall, специфичность** – доля всех показанных объектов относительно релевантных объектов.



## Метрики, учитывающие порядок рекомендаций

- Precision @ K - штрафует алгоритм за нерелевантные рекомендации среди топ-K первых
- AveragePrecision @ K - штрафует алгоритм за неправильный порядок рекомендаций
- Normalized Discounted Cumulative Gain (NDCG) – аналогично
- Интуиция: если при неверном порядке рекомендация на позициях 1 и 2 алгоритм наказывается сильнее, чем, для позиций 5 и 6.
- Мотивация – чем выше релевантность объекта – тем он должен быть выше.

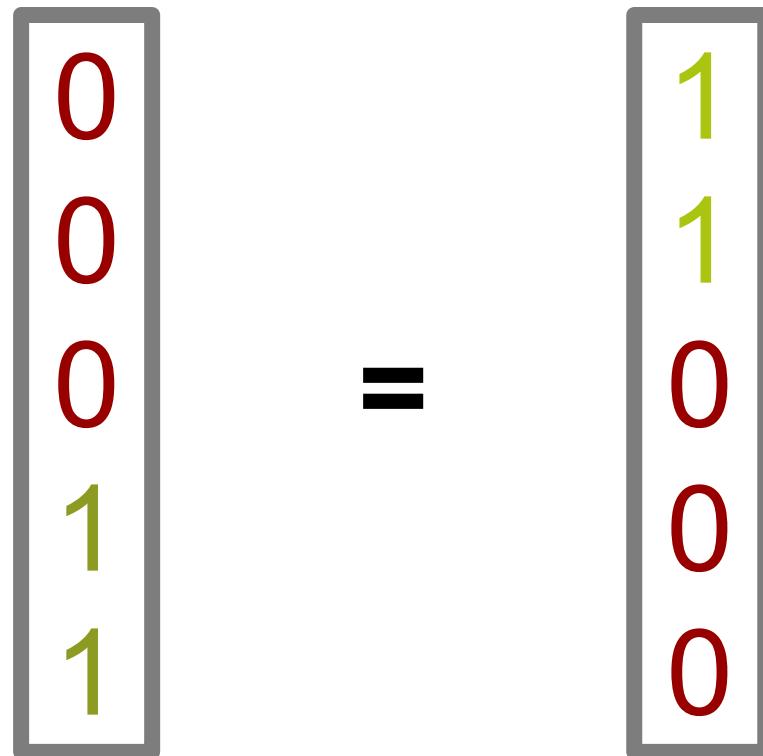
Если ошибаемся в хвосте рекомендаций – не так страшно.

## Precision @ K

Precision @ K – доля релевантных для пользователя объектов относительно первых K объектов, которые были ему показаны в соответствии с рангом рекомендации.

$$Precision@K = \frac{1}{K} \sum_{i=1}^K [y_{(i)} == 1]$$

где  $(y_{(1)}, \dots, y_{(K)})$  - **бинарные** метки релевантности предсказаний модели для запроса  $\mathbf{q}$  и полученных элементов  $\mathbf{d}$ , отсортированные по убыванию релевантности.



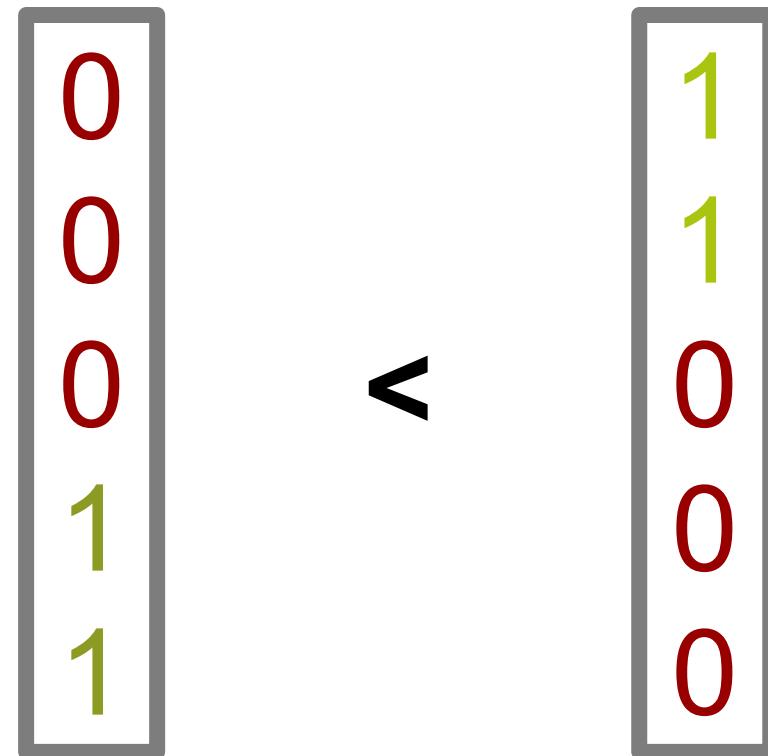
$$Precision@K = \frac{2}{5} = 0.4 \quad Precision@K = \frac{2}{5} = 0.4$$

## Average Precision @ K

**Average Precision @ K** – взвешенная кумулятивная версия **Precision @ K**

$$AP@K = \sum_{i=1}^K \frac{y_{(i)}}{\sum_{j=1}^K y_{(j)}} Precision@i$$

где  $(y_{(1)}, \dots, y_{(K)})$  - **бинарные** метки релевантности предсказаний модели для запроса **q** и полученных элементов **d**, отсортированные по убыванию релевантности.



$$\frac{1}{2} \cdot \frac{1}{4} + \frac{1}{2} \cdot \frac{2}{5} = \frac{13}{40} < \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 = 1$$

## Discounted Cumulative Gain (DCG)

**DCG @ K** – линейная комбинация функций прироста (gain) и штрафа (discount)

$$\text{DCG } @ K = \sum_{i=1}^K g(\tilde{y}_{(i)})d(i)$$

где  $(y_{(1)}, \dots, y_{(K)})$  - **вещественные** предсказания релевантности, отсортированные по убыванию

$\tilde{y}_{(i)}$  - истинные оценки релевантности

Примеры:       $g(a) = 2^a - 1$                    $d(b) = \frac{1}{\log(1 + b)}$

## **Normalized Discounted Cumulative Gain (DCG)**

**nDCG @ K** – нормированная версия **DCG @ K**:

$$\text{NDCG @ K} = \frac{\text{DCG @ K}}{\text{IDCG @ K}}$$

**IDCG @ K** = **Ideal DCG @ K** - получается, если отсортировать выдачу по истинным значениям релевантности:

$$\text{IDCG @ K} = \max(DCG@K)$$

## Дополнительные экзотические метрики

- Разнообразие
- Покрытие запроса
- Новизна
- Неожиданность

Если клиент купил молоко и хлеб можно не рекламировать ему яйца – он с высокой вероятностью их и так купит.

Нужно рекомендовать что-то с более низкой вероятностью покупки, но более разнообразное и интересное.

## Бизнес метрики

- Конверсия
- Клики
- Retention
- Прибыль
- LTV - life-time value

У рекомендательной системы много нематематических аспектов:  
место расположения, дизайн, и т.д.

Чтобы это корректно оценивать – нужны АВ тесты на пользователях.

АВ тесты часто единственный способ честно оценить бизнес ценность модели.

Sad but true: Такие метрики тяжело прогнозировать.

# Метрики



Лучше не использовать



Можно использовать



Наибольшая корреляция  
с бизнес метриками

## Основаны на ошибке

RMSE, MAE

## Основаны на релевантности

precision, recall

F1-score, accuracy

HR (hit rate)

## Основаны на ранжировании

nDCG (normalized discounted cumulative gain),

MRR (mean reciprocal rank),

AUCP (area under the TOPK-curve)

AUC (area under the ROC-curve)

## Измеряют другие аспекты:

- Покрытие (Coverage)
- Новизна (Novelty)
- Неожиданность (Serendipity)
- Разнообразие (Diversity)
- Доверие (Trust)
- Полезность (Utility)

Подробнее: “Evaluating Recommendation Systems”,  
Guy Shani and Asela Gunawardana, 2011

# Алгоритмы построения рекомендательных систем

# Схема построения рекомендательной системы

- Постановка задачи:

$U = \{u_1, \dots, u_M\}$  - множество пользователей

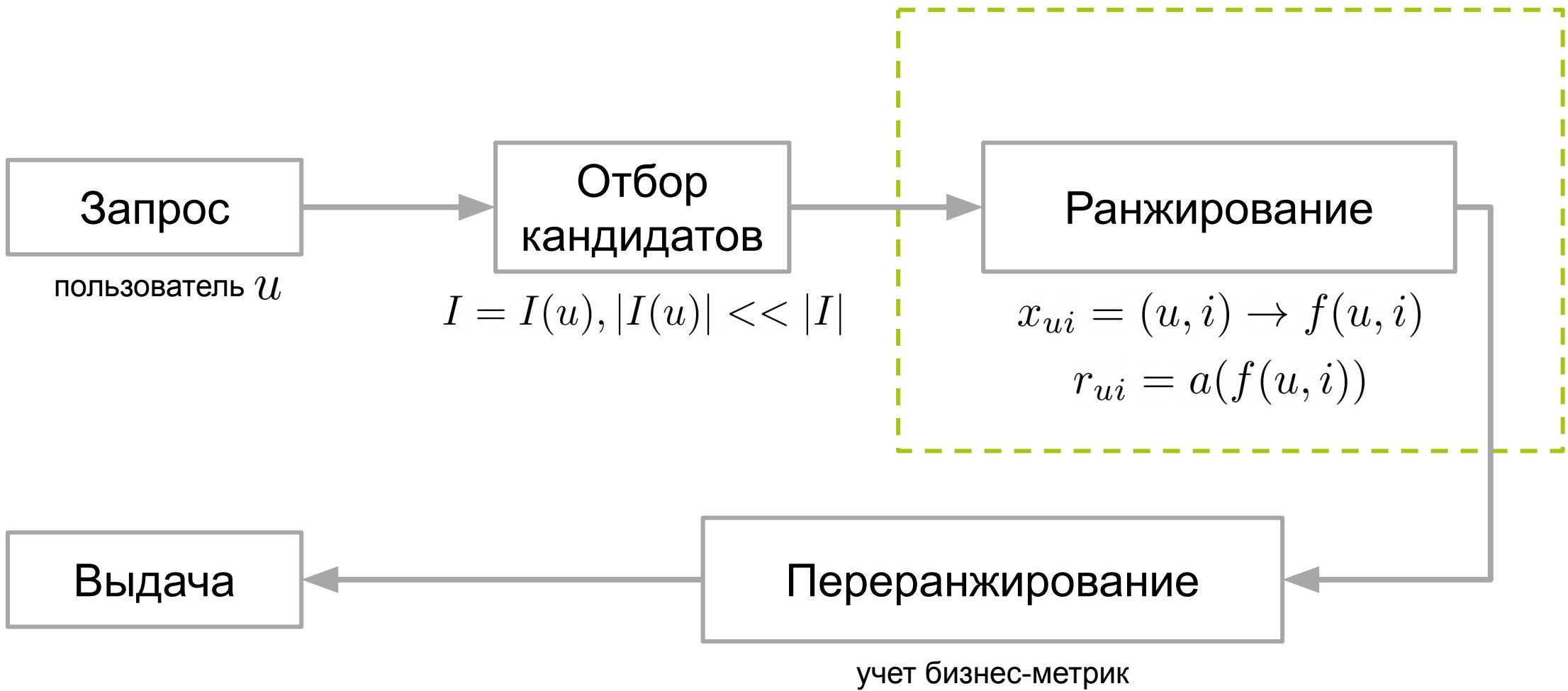
$I = \{i_1, \dots, i_N\}$  - множество элементов

- Для некоторых пар известны оценки релевантности:  $(u, i) : \exists r_{ui} \in \mathbb{Y}$

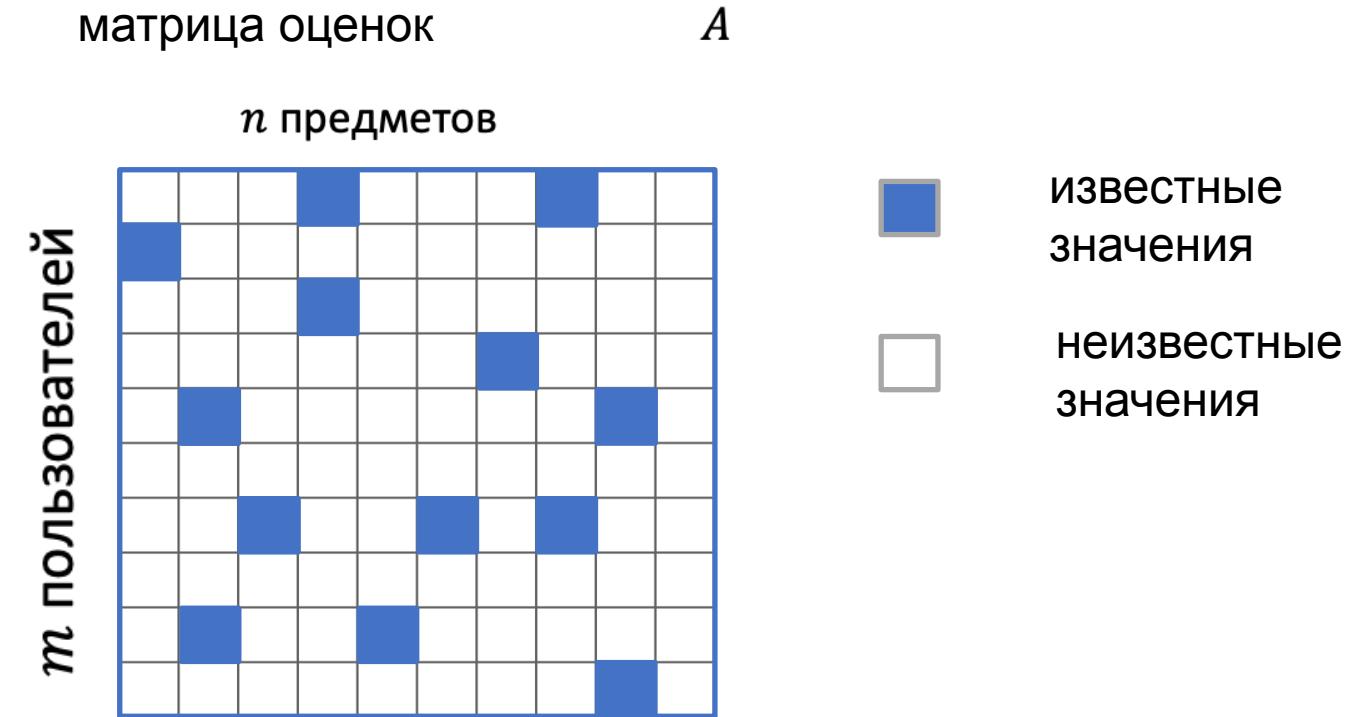
$R = \{(u, i) | \exists r_{ui}\}$  - множество оценок

- Требуется: предсказать  $a(u, i) = r_{ui} \quad \forall u \in U, i \in I$

# Схема построения рекомендательной системы



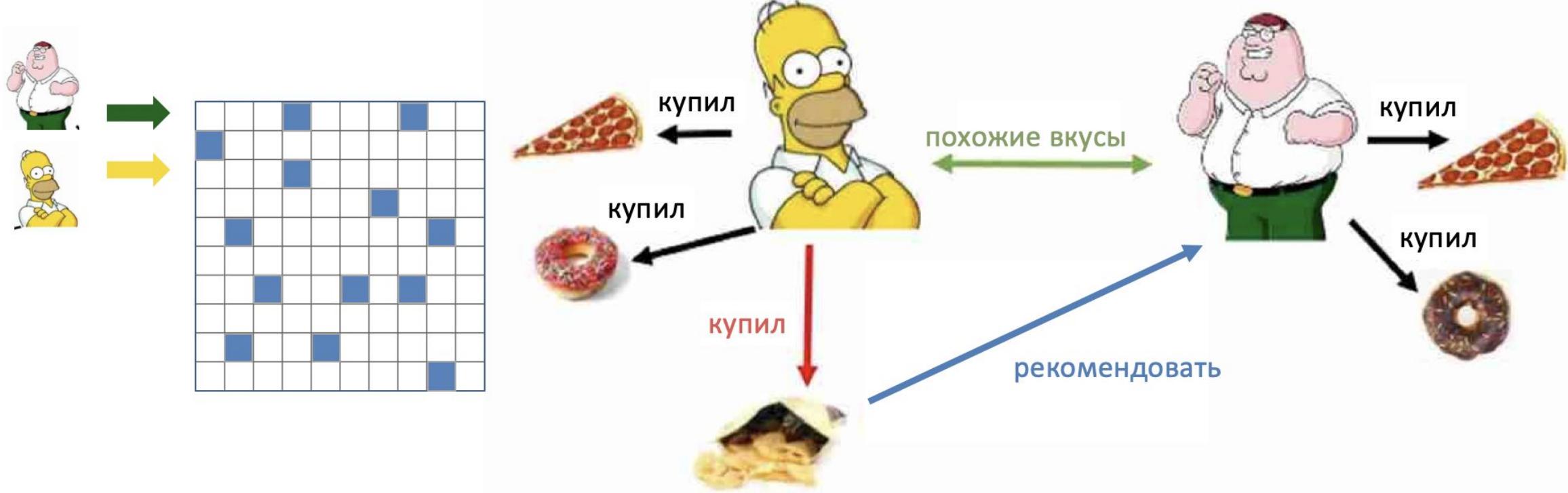
# Данные для построения рекомендательной системы



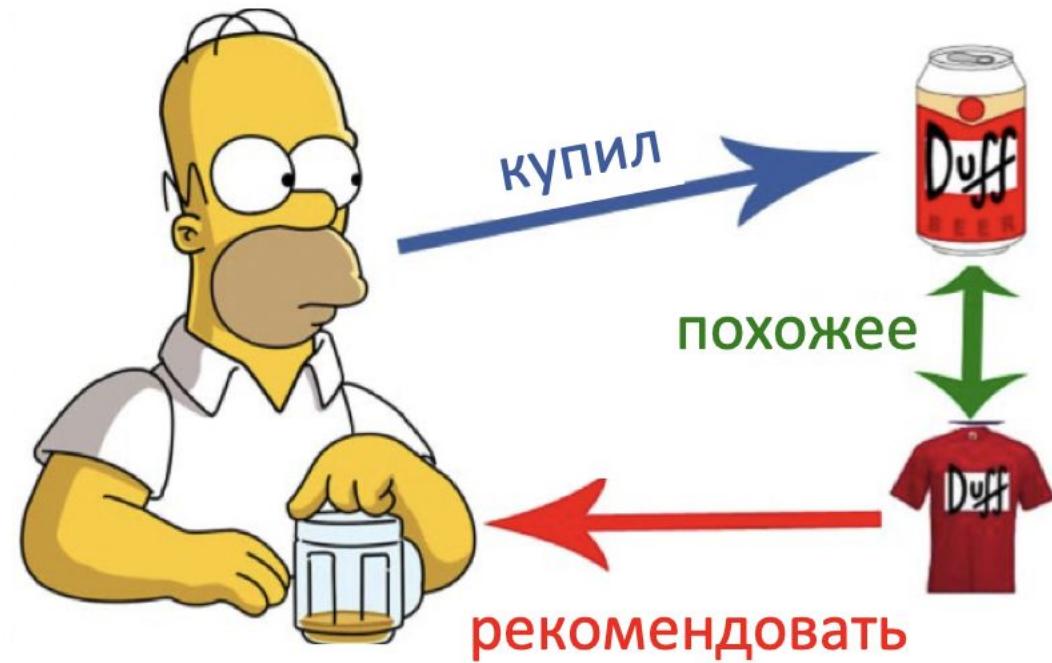
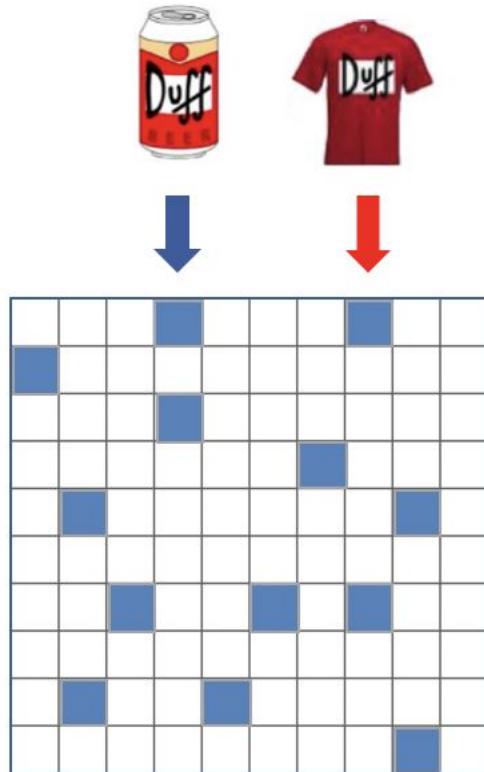
**Задача:** найти функцию полезности  $f_U$ , генерирующую предсказания для любой пары:

$f_U$ : пользователь + предмет  $\rightarrow$  релевантность

**Если мы умеем считать расстояния между пользователями, то можем использовать коллаборативную фильтрацию по схожести пользователей (*user-based*)**

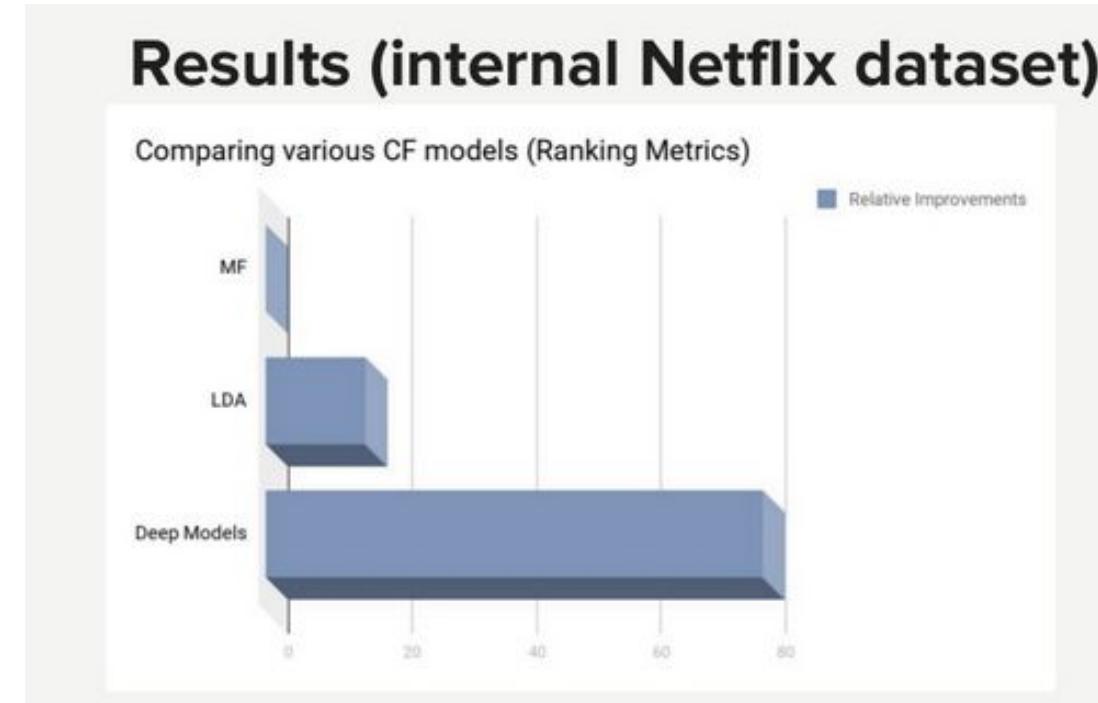


**Если мы умеем считать расстояния между предметами, то можем использовать коллаборативную фильтрацию по схожести предметов (*item-based*)**



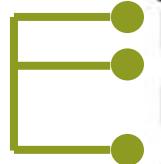
# Немного о более сложных методах

Как пишут в статьях



Как все работает на практике

Матричная  
факторизация  
у 3 из 4 топовых  
алгоритмов



Method	movielens			yahoo			pinterest		
	HR [%]	ARHR [%]	NDCG [%]	HR [%]	ARHR [%]	NDCG [%]	HR [%]	ARHR [%]	NDCG [%]
EigenRec	45.21	20.44	26.35	48.12	23.30	29.23	33.81	13.51	18.41
PureSVD	44.14	19.33	25.36	38.68	18.30	22.62	30.97	11.85	16.30
RP3b	34.87	15.02	19.66	41.51	17.82	22.94	27.01	8.07	12.45
SLIM	46.34	21.39	27.28	52.44	26.15	32.35	34.17	13.63	18.57
Mult-DAE	44.06	18.97	24.83	45.37	21.46	27.07	35.03	13.79	18.77
Mult-VAE	44.35	19.50	25.31	45.09	21.22	26.80	35.13	13.73	18.71

# Персонализация контента – трудная задача

## Пропуски в данных

- 99.99...% неизвестных значений
- смещенность оценки, Missing Not at Random (MNAR)

## «длинный хвост»

- 5% предметов могут покрывать 40% всех взаимодействий
- нишевые рекомендации

## оценка качества

- Выбор метрики качества, нестандартные метрики (novelty, diversity...)
- проверка качества оффлайн vs. A/B тестирование

## холодный старт

- неопределенность в том, что рекомендовать
- сложно подобрать наиболее репрезентативные рекомендации

## усложнение моделей

- включение признаковой информации
- включение контекста взаимодействия

## вычислительная скорость

- вычисление модели за разумное время
- выдача рекомендаций в реальном времени

# Холодный старт

Проблема всех рекомендательных систем:

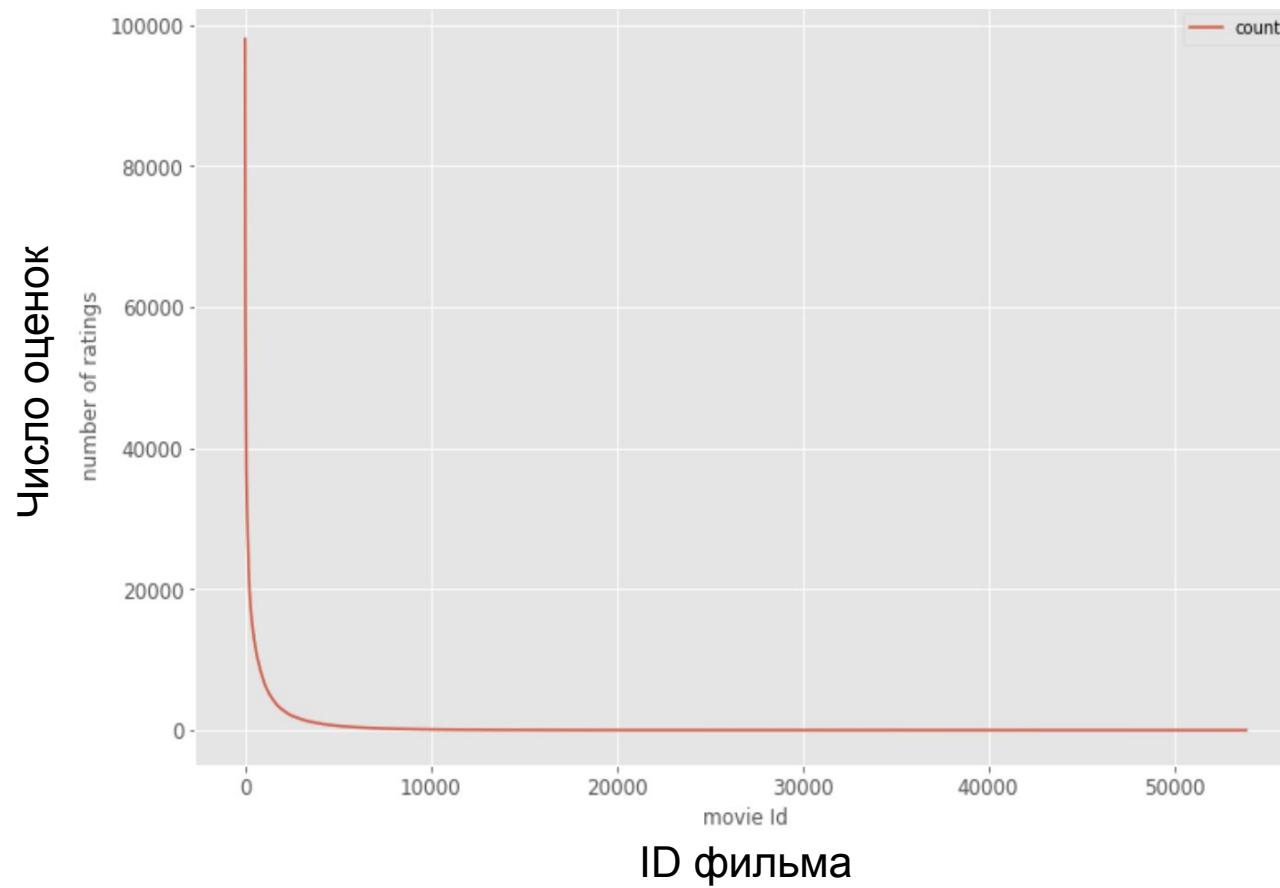
Первое время нет информации о предпочтениях пользователя.

Решения:

- Использовать профиль «среднего» пользователя
- Рекомендовать самые популярные товары

# Длинный хвост

Популярные товары покупают часто.  
Большинство товаров покупают редко.



Замкнутый круг: про длинный хвост мало статистики – система его не рекомендует – мало статистики.

Отсутствует новизна предложений.  
Гибридные подходы.

Решение:

Продуктовое – создавать категории новинок, иногда рекомендовать 1-2 случайных товара из хвоста.

Гибридные подходы:

Ансамбли из рекомендательных алгоритмов.  
Минусы: долго вычисляется, сложно поддерживать

# Дополнительные материалы

## Кратко и на русском:

- Серия постов С. Николенко на habr.com
- “Как работают рекомендательные системы”, М. Ройзнер, <https://habr.com/ru/company/yandex/blog/241455/>
- Анатомия рекомендательных систем:
  - Часть 1: <https://habr.com/ru/company/lanit/blog/420499/>
  - Часть 2: <https://habr.com/ru/company/lanit/blog/421401/>

## Кейсы:

- Что происходит в мире рекомендательных систем прямо сейчас <https://youtu.be/8bqgkVmXi0>
- Рекомендации в Okko <https://habr.com/ru/company/okko/blog/454224/>
- Как рекомендовать музыку, Яндекс <https://habr.com/ru/company/yandex/blog/441586/>
- 10 уроков рекомендательной системы Quora <https://habr.com/ru/company/retailrocket/blog/341346/>

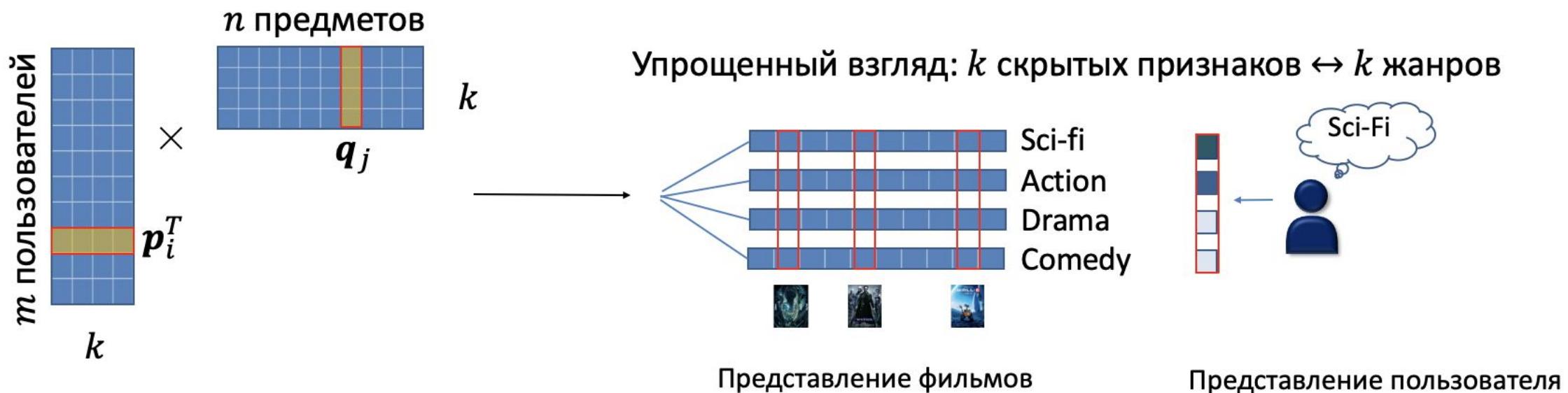
## Книжки:

- Recommender Systems Handbook, 2015, 2<sup>nd</sup> edition; F. Ricci, L. Rokach, B. Shapira
- Recommender Systems. The Textbook, 2016; Charu C. Aggarwal
- Recommender Systems: An Introduction, 2010; D.Jannach, M.Zanker, A.Felfernig, G.Friedrich
- Collaborative Recommendations: Algorithms, Practical Challenges and Applications; S. Berkovsky, I. Cantador and D. Tikk.

# Матричная факторизация

# Представление пользователей и предметов – способ подсчитать релевантность

Мы построили представление размерности  $r$  для пользователей и предметов



## Релевантность – расстояние между пользователями и предметами

$$r_{ij} \approx \mathbf{p}_i^T \mathbf{q}_j = \sum_{t=1}^k p_{it} q_{jt}$$

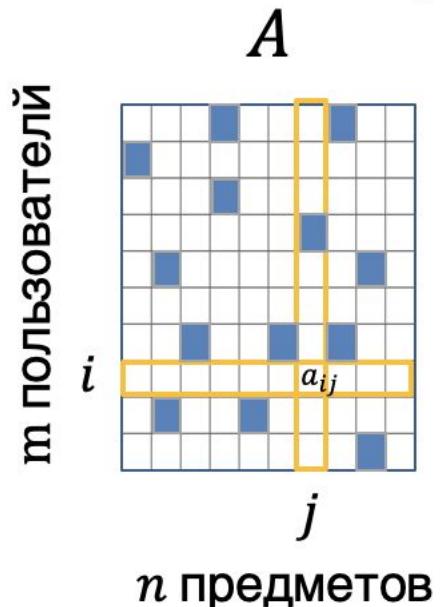
$\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{ik})$  – вектор скрытых признаков пользователя  $i$   
 $\mathbf{q}_j = (q_{j1}, q_{j2}, \dots, q_{jk})$  – вектор скрытых признаков предмета  $j$

# Как получить представления пользователей: матричная факторизация

Основное предположение:

относительно небольшое количество шаблонов  
поведения + небольшие вариации за счет  
индивидуальных особенностей

$$\mathcal{L}(A, P, Q) \rightarrow \min_{P, Q}$$



$$\mathcal{L}(A, P, Q) = \|A - PQ^T\|_2$$

Строки матриц  $P$  и  $Q$  дают представление пользователей и предметов на основе набора скрытых признаков

$$a_{ij} \approx r_{ij} = \mathbf{p}_i^T \mathbf{q}_j = \sum_{t=1}^k p_{it} q_{jt}$$

Релевантности должны быть похожи на элементы исходной матрицы  $A$

# Дополнительные слайды

# Фреймворки и библиотеки

## Frameworks

Surprise

<https://github.com/NicolasHug/Surprise>

Polara

<https://github.com/evfro/polara>

Collaborative Filtering – Apache Spark

<http://spark.apache.org/docs/latest/mllib-collaborative-filtering.html>  
(Neighborhood models and MF)

Turi Create (acquired by Apple)

<https://turi.com/learn/userguide/recommender/introduction.html>

Microsoft Azure

<https://github.com/Microsoft/Recommenders>

## Полезные библиотеки

Collaborative Filtering for Implicit Feedback Datasets

<https://github.com/benfred/implicit> (fastest)

<https://github.com/quora/qmf> (by Quora)

<https://github.com/MrChrisJohnson/logistic-mf> (as in Spotify)

Factorization Machines

<https://github.com/srendle/libfm>

## Другие библиотеки

Нейронные сети

<https://github.com/maciejkula/spotlight>

<https://github.com/MrChrisJohnson/deep-mf>

<https://github.com/songgc/TF-recomm>

<https://github.com/Netflix/vectorflow> (by Netflix)

Билинейные модели

<https://github.com/lyst/lightfm/>

<http://www.recsyswiki.com/wiki/SVDFeature>

MyMediaLite (used to be popular)

<http://www.my medialite.net>

Модели с латентными факторами

<https://github.com/zhangsi/CisRec>

Простой движок для рекомендаций на основе контента  
<https://github.com/groveco/content-engine>

Логистическая матричная факторизация

<https://github.com/MrChrisJohnson/implicit-mf>

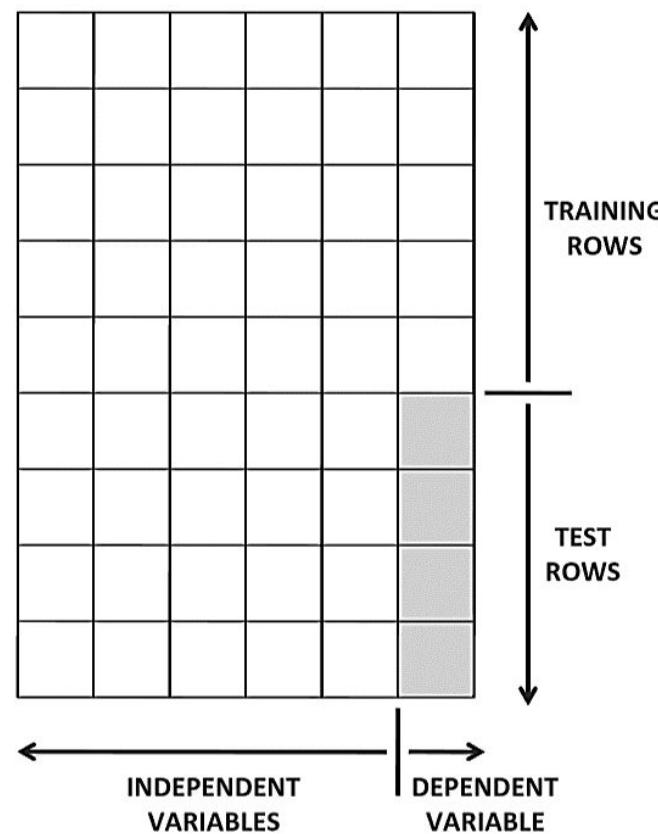
Hermes (поддерживает Spark)

<https://github.com/Lab41/hermes>

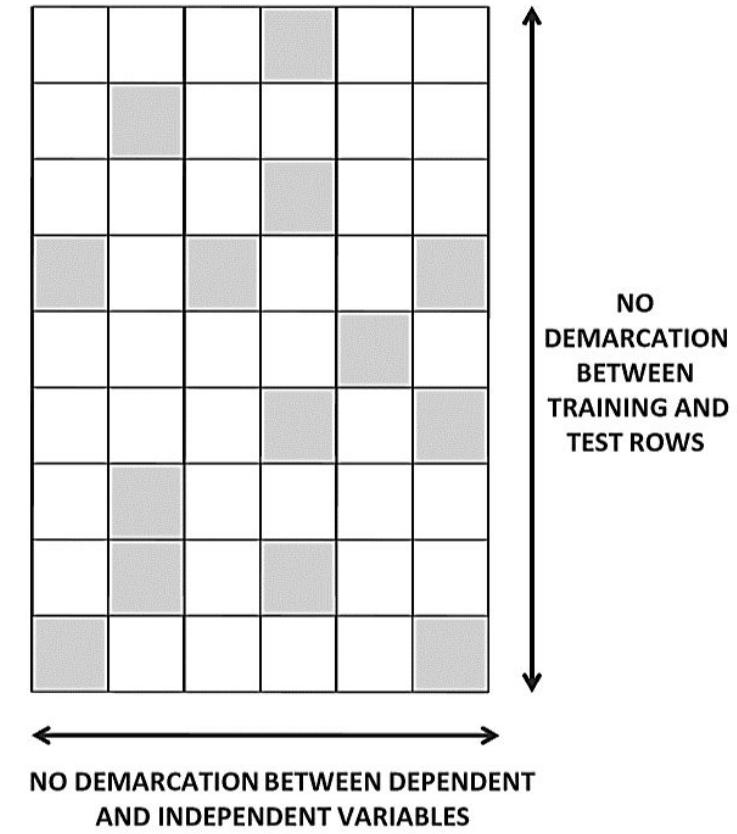
# Задачи классификации/регрессии vs задача предсказания предпочтений

Классов в задаче классификации –  
немного

Объектов в рекомендательных  
системах – очень много

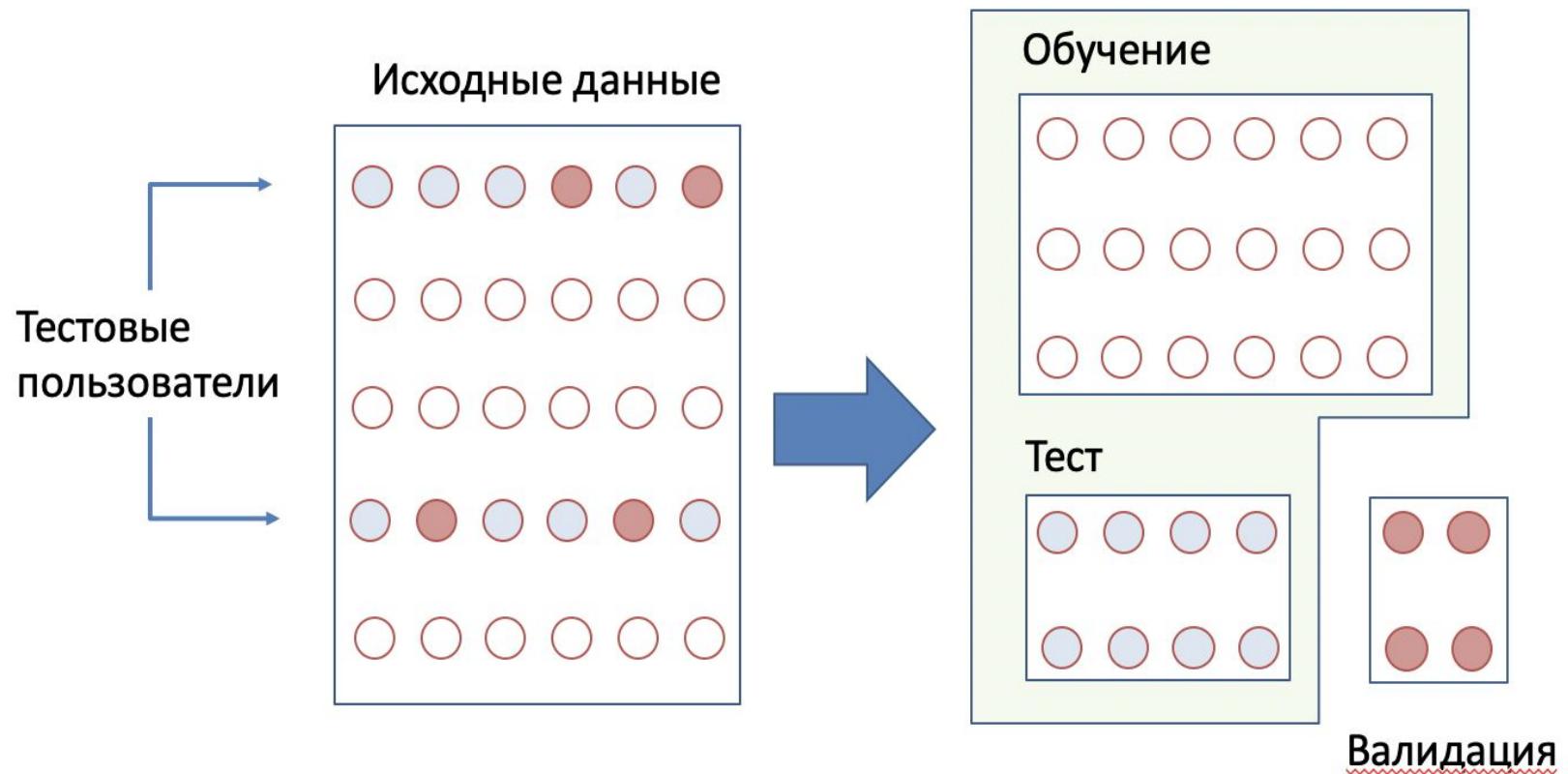


Структура данных  
в задаче классификации



Структура данных  
в рекомендательных  
системах

# Разбиение на обучающую и тестовую выборку



Данные разбиваются по пользователям, а не взаимодействиям.

Обычно Обучение = Наблюдения  $\cup$  Тест.

Соотношение: 80/20.

Валидация может содержать топовые или случайные объекты.

Тест и валидация – те же пользователи.

**Машинное обучение**

**продает клиентам продукты**





## Премиальная кредитная карта Mastercard

- Есть продукт
- Нам нужно понять, каким клиентам нужно рекомендовать
- Делаем отдельную модель для рекомендации продукта

## Фильмы для вас >



- Есть миллионы продуктов (объектов)
- Нам нужно понять, какие из них скорее всего понравятся клиенту
- Отдельных моделей слишком много
- Основные данные - взаимодействия

**Не** рекомендательная система,  
задача классификации

Рекомендательная система

# Предсказание интереса к продукту

1. Прогнозируем, что клиент купит продукт по истории клиента и его общим характеристикам
2. Доступны данные за несколько последних лет
3. Строим модель градиентного бустинга
4. Максимизируем ROC AUC



# Уровни аналитики для анализа ухода клиентов

Дескриптивная аналитика:  
что произошло?

- Какие клиенты купили?
- Сколько мы заработали денег?
- Почему они купили?

Предиктивная аналитика:  
что произойдет?

- Какие клиенты купят?
- Сколько мы заработкаем денег?
- Почему они купят?

Прескриптивная аналитика:  
что нам с этим делать?

- Как повлиять на клиента, чтобы он купил?
- Когда имеет смысл влиять с точки зрения оптимизации ресурсов?

# Предсказание интереса к продукту не решает бизнес задачу

Предсказание покупки

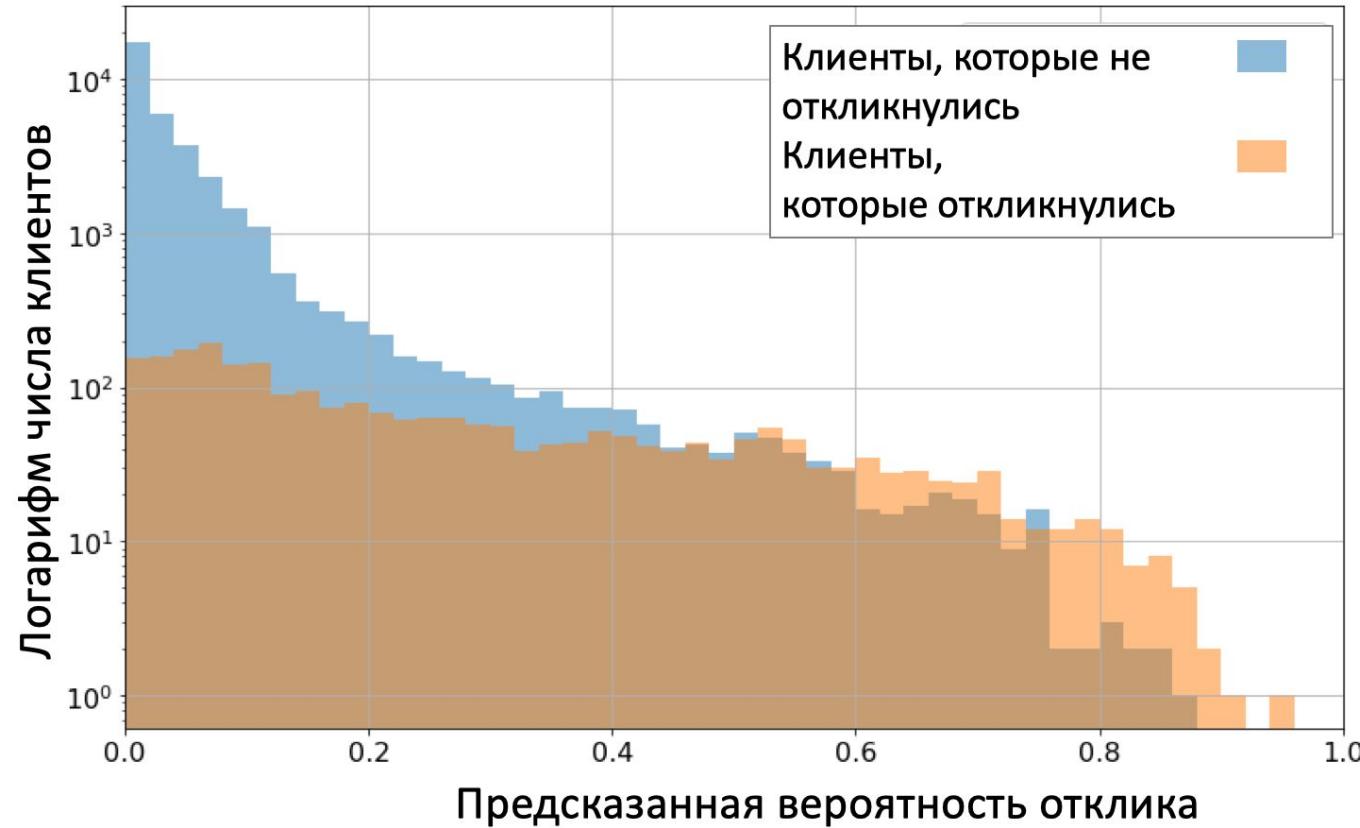
Какие клиенты купят?

Предсказание  
правильного  
действия

На каких клиентов повлиять,  
чтобы они купили продукт?

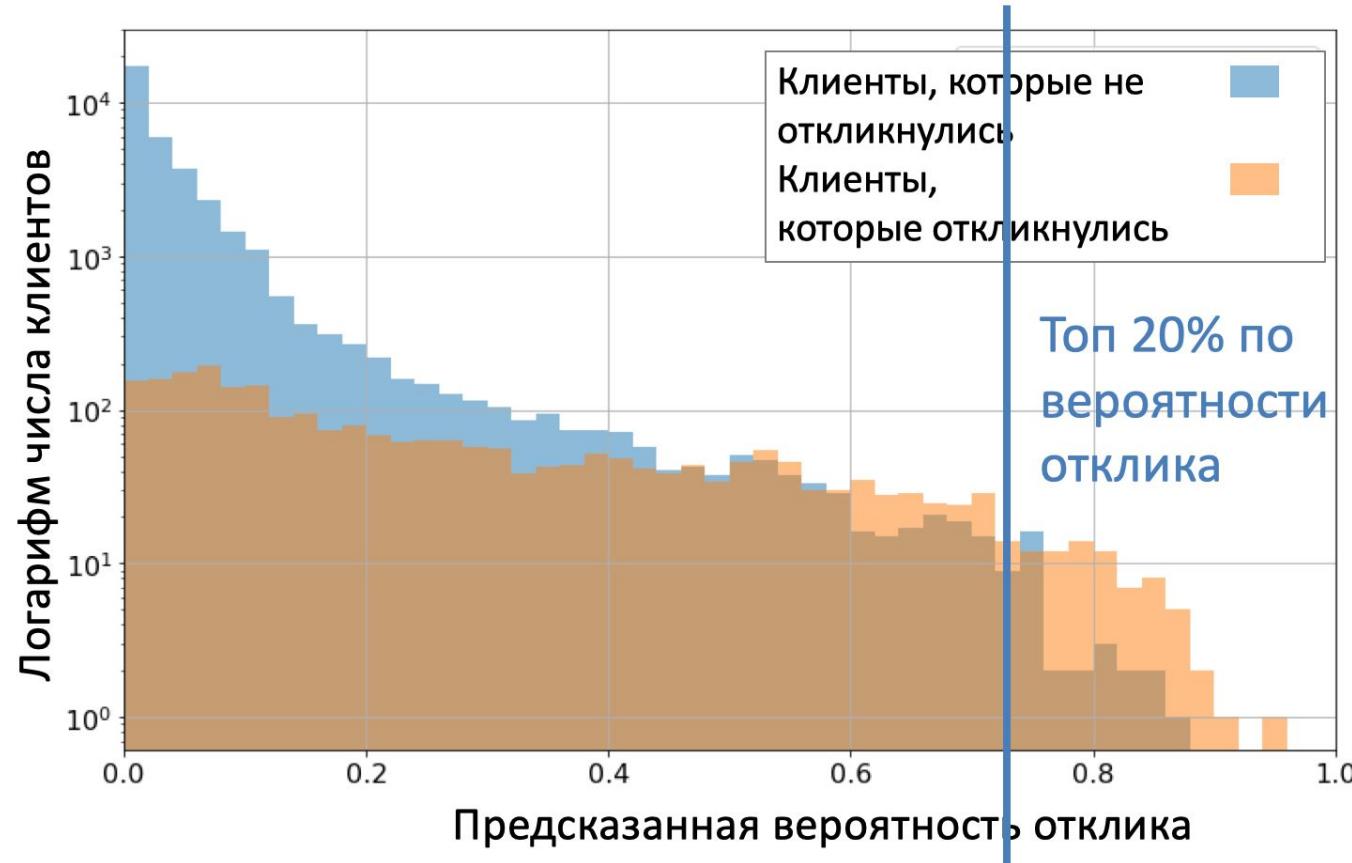
# Выход модели классификации – вероятность отклика

Гистограмма вероятности отклика для клиентов из тестовой выборки



# Выбирая порог, смотрим на процент откликов в каждом 10% по вероятности отклика

Гистограмма вероятности отклика для клиентов из тестовой выборки



# Кейс использования uplift моделирования: таргетируем предложение для ритейла

Тестовая группа – сделали рассылку sms, Контрольная группа – ничего не рассыпали

Дециль клиентов по вероятностям отклика	Процент откликов для тестовой группы	Процент откликов для контрольной группы	Прирост
1 (топ 10%)	26.99%	27.90%	-0.91%
2 (от 10% до 20%)	20.34%	20.90%	-0.56%
3 (от 20% до 30%)	10.70%	10.04%	0.66%
4 (от 30% до 40%)	8.90%	7.52%	1.38%
Среднее по 4 декилиям	16.73%	16.59%	0.14%

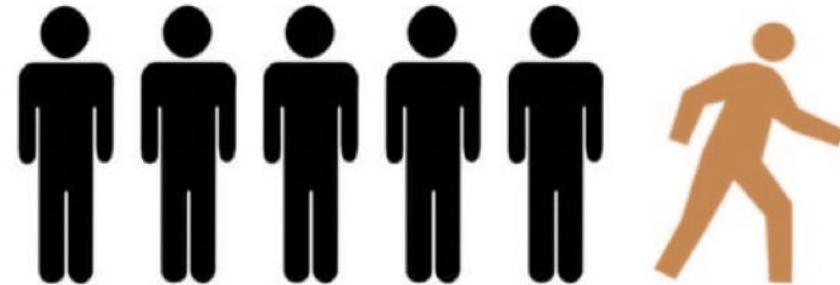
<https://www.predictiveanalyticsworld.com/machinelearningtimes/uplift-modeling-making-predictive-models-actionable/8578/>

# Предсказание оттока клиента как задача машинного обучения

1. Прогнозируем уход (нет действий за три месяца) по истории клиента и его общим характеристикам
1. Доступны данные за несколько последних лет
1. Строим модель градиентного бустинга
1. Максимизируем ROC AUC или F1



**Премиальная кредитная карта Mastercard**



# Предсказание оттока клиента не решает бизнес задачу

Предсказание оттока  
клиента

Какие клиенты уйдут?

Предсказание  
правильного  
действия

Как повлиять на клиента,  
чтобы он от нас не ушел?

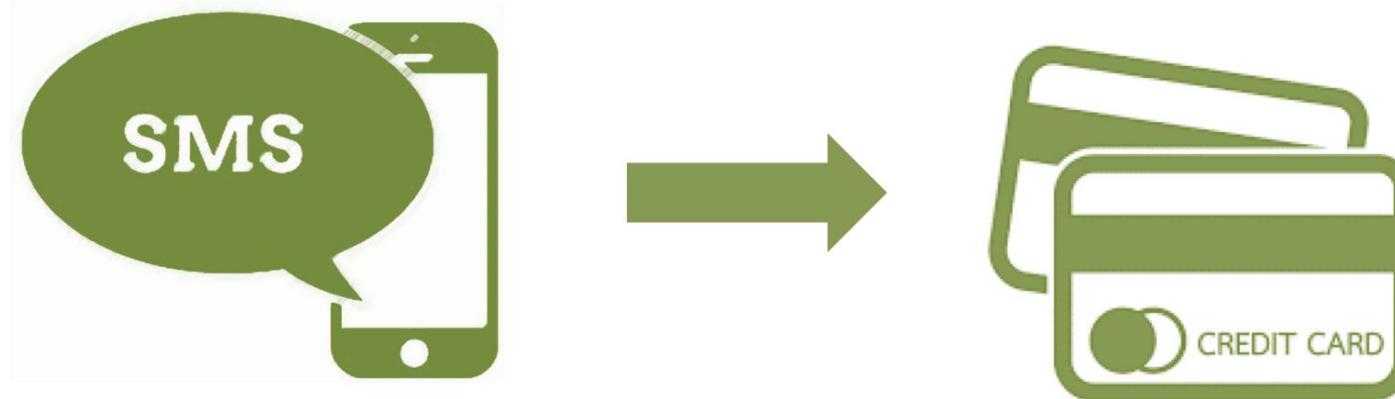
# Что мы хотим предсказывать, чтобы решить бизнес-задачу:

Каким клиентам нужно предложить скидку, чтобы они от нас не ушли?

Каким клиентам нужно по SMS рассказать про новый продукт, чтобы они купили этот продукт?

# Продвижение продукта с помощью машинного обучения

Хотим построить модель машинного обучения, которая будет нам говорить, следует ли контактировать с клиентом или нет, предлагая ему продукт банка



# Есть 4 типа клиентов, нам нужны только клиенты типа **Убеждаемые**



# Продвижение продукта с помощью машинного обучения

## Обучающая выборка

- *Look-alike* модель – вероятность того, что клиент выполнит действие
  - известные позитивные объекты
  - случайные негативные объекты
- *Response* модель – вероятность того, что клиент выполнит действие при условии контакта
  - данные, собранные в ходе контактов
- *Uplift* модель – разница между вероятностями действий с контактом и без
  - **непонятно, сейчас подумаем**

# Uplift модель

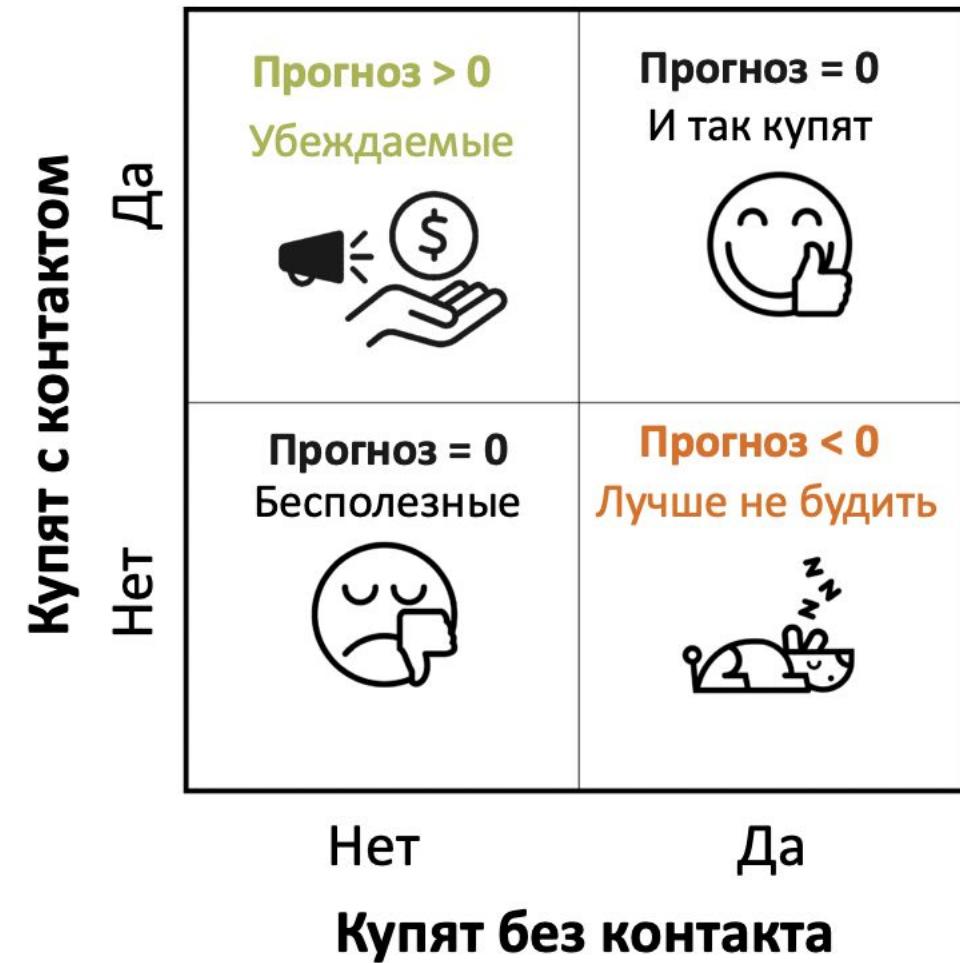
Прогноз Uplift модели

=

Вероятность того, что  
клиент купит, если у него был контакт

–

Вероятность того, что  
клиент купит, если у него не было контакта



# Uplift модель: как обучать?

Нельзя одновременно послать и не послать смс!

Соберем выборку, используя следующий план эксперимента:

1. Разобьем на целевую и контрольную группу все наше множество клиентов
2. Запустим пилот кампании на целевую группу



# Uplift модель с признаком коммуникации: как обучать?

Обычные входные признаки	Бинарный признак коммуникации	Целевая переменная
Пол, возраст, город проживания	0 0 1 0 1 ...	1 0 0 0 1 ...

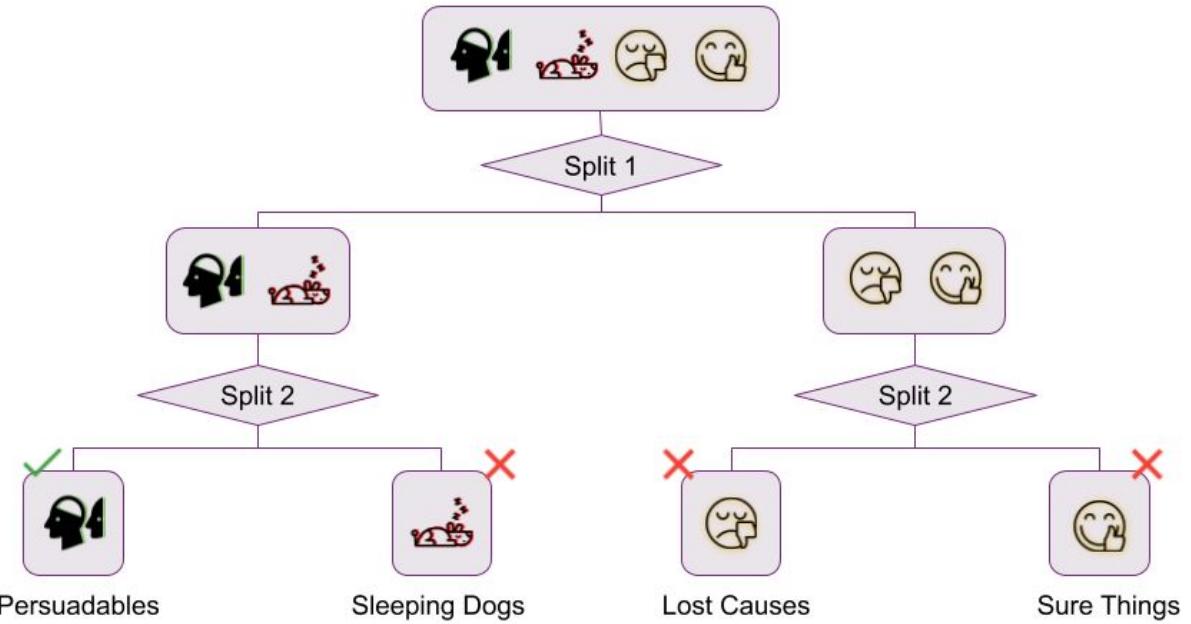


Внутренняя модель

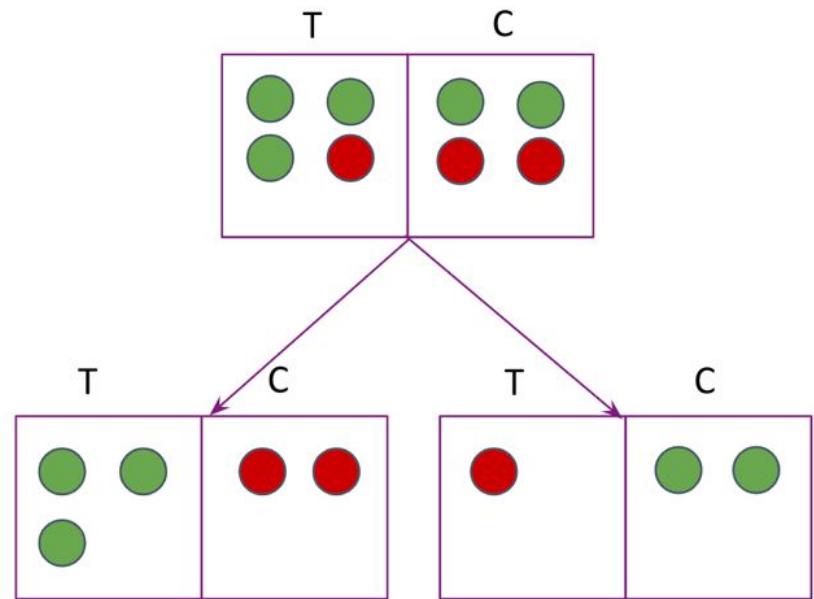
# Uplift модель с признаком коммуникации: как обучать?



# Uplift деревья



В каждом узле делим выборку так, чтобы пользователи в целевой и контрольной группе отличались как можно сильнее



# Таргетируем предложение для ритейла

Дециль клиентов по вероятностям отклика	Процент откликов для тестовой группы	Процент откликов для контрольной группы	Прирост
1	26.99%	27.90%	-0.91%
2	20.34%	20.90%	-0.56%
3	10.70%	10.04%	0.66%
4	8.90%	7.52%	1.38%
Среднее по 4 децилям	16.73%	16.59%	0.14%

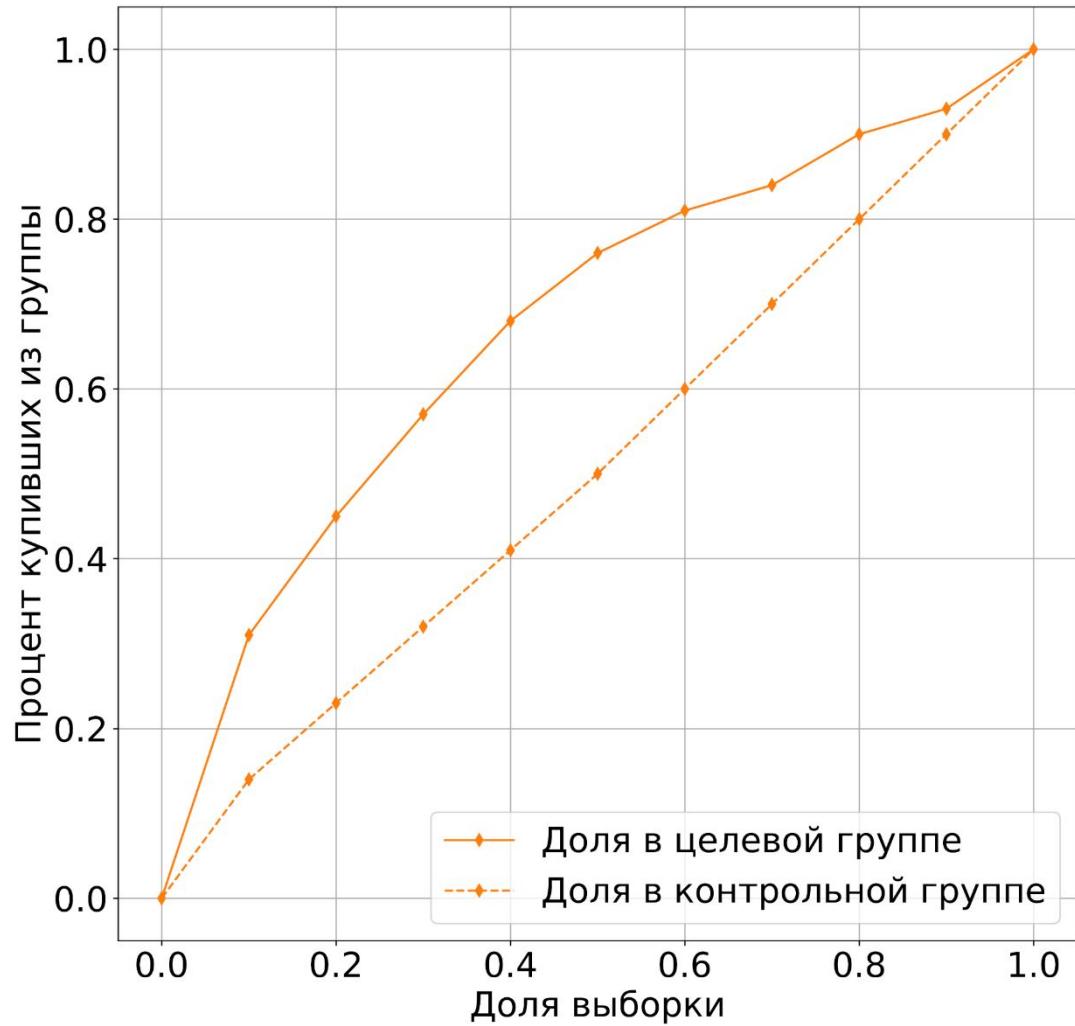
Модель предсказывает отклик клиента

Дециль клиентов по uplift	Процент откликов для тестовой группы	Процент откликов для контрольной группы	Прирост
1	18.80%	12.90%	5.90%
2	7.80%	5.40%	2.40%
3	6.90%	4.50%	2.50%
4	4.30%	3.60%	0.70%
Среднее по 4 децилям	9.45%	6.60%	2.88%

Модель предсказывает uplift клиента

**Uplift** – разность между вероятностью покупки с коммуникацией и вероятностью покупки без коммуникации

# Сравним выделенный процент купивших клиентов для тестовой и контрольной группы

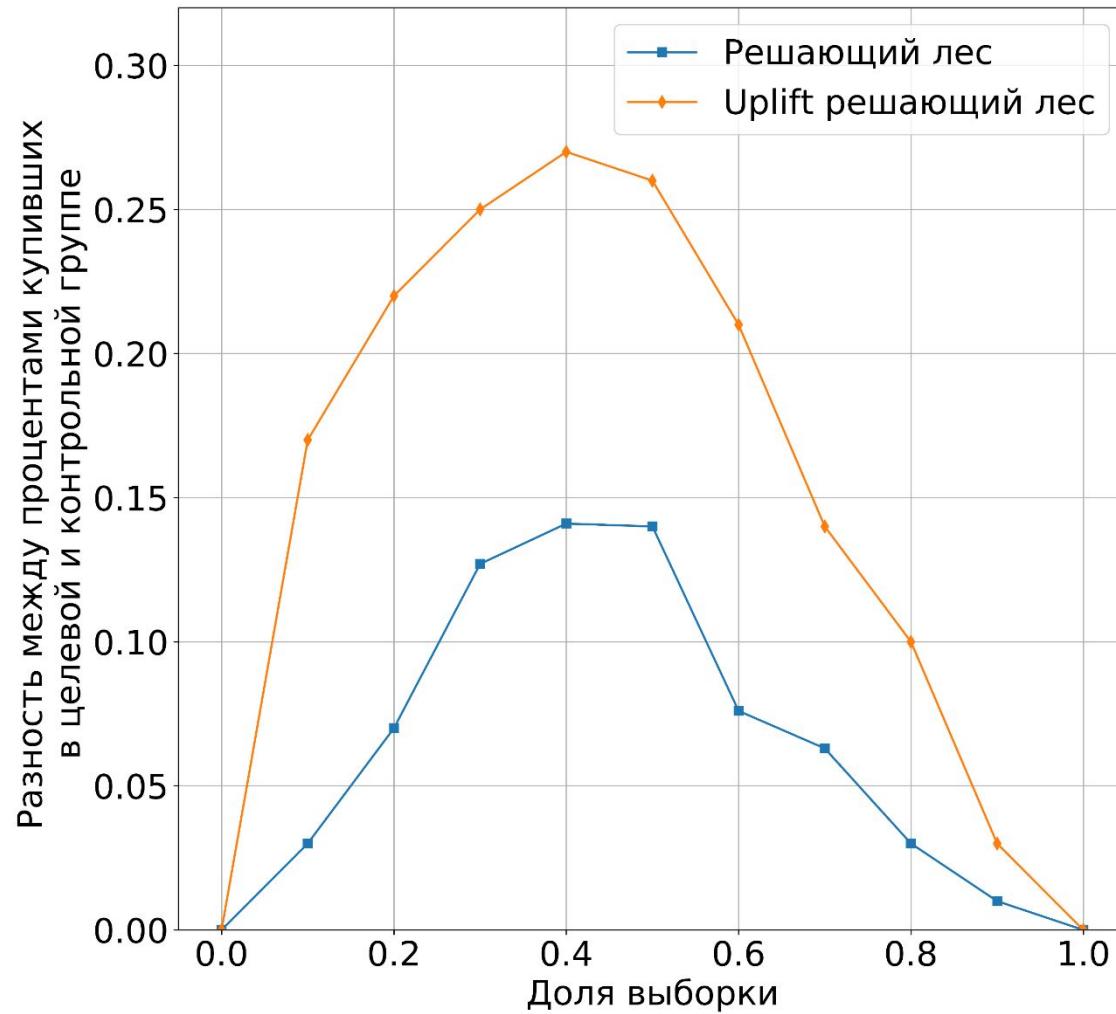


**Доля в тестовой группе** – какой процент откликнувшихся мы выделили из тестовой группы

**Доля в контрольной группе** – какой процент откликнувшихся мы выделили из контрольной группы

Если мы посчитаем разность между этимиолям – получим прирост и **lift curve**

# Lift curve как показатель качества работы uplift модели



Так как у Uplift решающего леса разность выше, эта модель лучше выделяет тех, кому нужно продавать продукт

# Зачем еще Uplift модели

- Понимаем, кому можно предложить повышенную ставку по кредиту – и клиент на это согласится
- Предсказываем, кому давать лекарство
- Предсказываем, на какой товар ставить акцию



# Дополнительные слайды

# В реальной жизни нужно учитывать и другие эффекты

**Каннибализация** – сокращение спроса одного продукта за счет возросшего спроса на другой продукт в аналогичной категории.

**Гало** – продажи сопутствующего товара растут, если выросли продажи исходного товара

Можно закладывать в модель, так как для нее не очевидны зависимости спроса двух схожих товаров.

Модель зависит от количества продуктов:

- Для 10-20 продуктов можно настроить матрицу каннибализации вручную
- Для большого числа продуктов универсального решения нет