



人工智能教育

# JULYEDU

## 推荐系统实战之 用户建模

讲师: Fred

<https://www.julyedu.com/>

# 目录

- 为什么需要用户建模
- 用户人口学属性
- 文本挖掘
- 权重计算
- 实战:倒排索引/人口学属性预测

今日头条User Profile系统架构实践

原创：丁海峰 QCon 2016-08-10



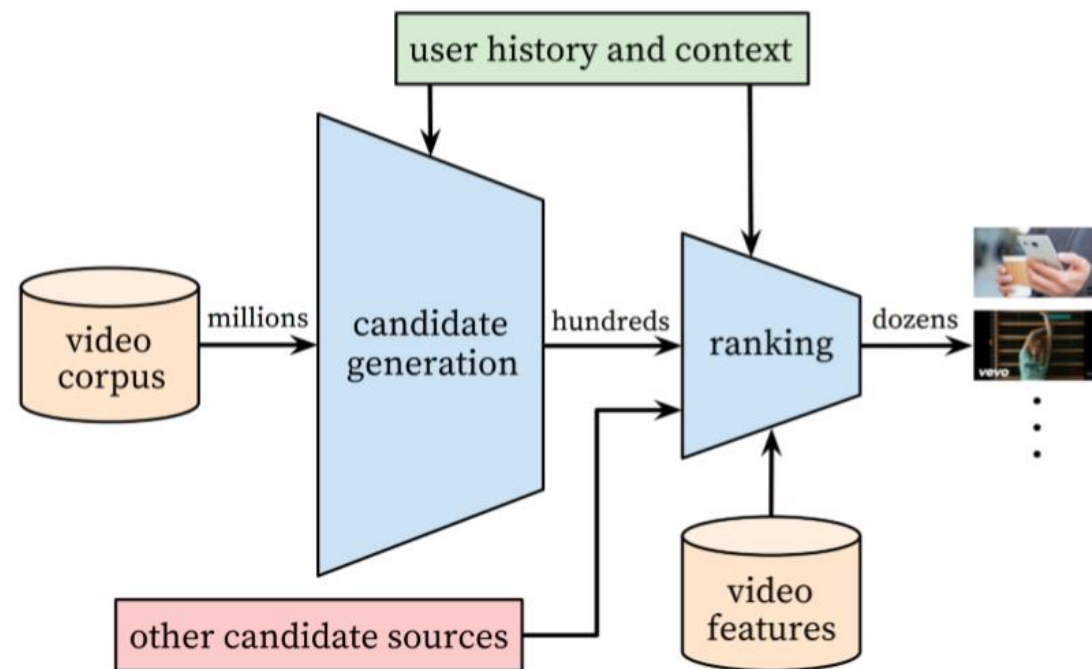
[https://mp.weixin.qq.com/s?\\_\\_biz=MzU3OTgyMDAwNw==&mid=2247488828&idx=1&sn=2febdd5a7090c9533592728bd8d5889f&chksm=fd6115b4ca169ca2336f1cc00218506a919e87a9a2862ebfc3f7d00aa71dcffde744c1bc30bc&mpshare=1&scene=23&srcid=](https://mp.weixin.qq.com/s?__biz=MzU3OTgyMDAwNw==&mid=2247488828&idx=1&sn=2febdd5a7090c9533592728bd8d5889f&chksm=fd6115b4ca169ca2336f1cc00218506a919e87a9a2862ebfc3f7d00aa71dcffde744c1bc30bc&mpshare=1&scene=23&srcid=)



# 01

## 为什么需要用户建模

# 推荐系统架构



# 用户画像(User Profile)

- 推荐系统中最核心的数据之一是 user profile 数据。需要从大量历史用户行为中分析和挖掘各种维度的特征，来刻画用户的兴趣偏好。
- 需要怎么的用户画像？
  - 人口学
    - 性别、年龄、地域etc
  - 内容特征（标签）：category、topic、keyword、entity
    - 喜欢、不喜欢
    - 长期、短期
  - 协同特征
    - 相似用户



用户基本信息			
性别	展开>>	年龄段	展开>>
male	0.9452	24-30	0.3068
用户订阅来源			
越玩越野	2015-03-31 20:05	麻省理工	
图虫人像摄影	2015-01-19 14:15	图虫人文主	
枪妹党	2014-11-06 13:20	什么值得买	
兴趣分类来源(long term) @召回			
科技:36氪	1080.8857	科技:CSDN	
科技:虎嗅网	537.5638	科技:DoNews	
科技:虎嗅	417.8447	科技:界面	
科技:IDoNews	371.3731	科技:人人网	
国内:新京报	340.3811	财经:36氪	
用户喜欢分类信息:			
用户喜欢顶级分类(impr decay) @召回			
news_tech	1250.8073	news_world	389.5
news_travel	270.3227	news_society	255.4
news_military	107.3003	digital	74.35
兴趣汽车品牌 @召回			
丰田汽车	288.9407	大众汽车	144.20
日产汽车	56.7518	福特汽车	47.662
本田技研工业	24.0639	起亚汽车	22.765
现代汽车	17.3621	路虎	15.920
兴趣汽车价格 @召回			
35-50万	630.7637	25-35万	541.1
70-100万	217.4961	15-20万	120.0
5万以下	7.7931		

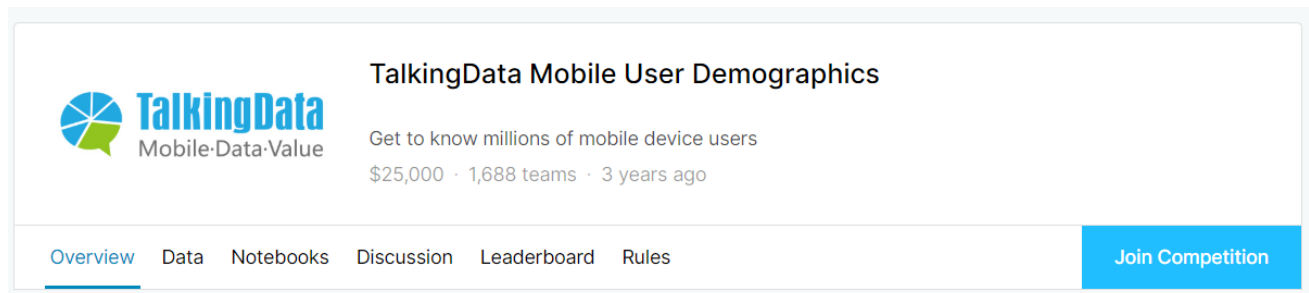


# 02

## 用户人口学属性

# 用户人口学属性

- 性别/年龄
  - 账号注册
  - 人工标注数据+规则+模型
  - 用户APP安装列表/用户行为
  - 手机品牌
- 地域
  - GPS定位
- Kaggle比赛
  - <https://www.kaggle.com/dvasyukova/a-linear-model-on-apps-and-labels>



The screenshot shows the top section of a Kaggle competition page. On the left is the TalkingData logo with the tagline 'Mobile·Data·Value'. To the right, the competition title 'TalkingData Mobile User Demographics' is displayed, followed by a description 'Get to know millions of mobile device users' and statistics '\$25,000 · 1,688 teams · 3 years ago'. Below this is a navigation bar with links: Overview (highlighted), Data, Notebooks, Discussion, Leaderboard, and Rules. A blue 'Join Competition' button is on the far right.

## Data Sources

app\_events.csv

app\_labels.csv

events.csv

gender\_age\_test.csv

gender\_age\_train.csv

label\_categories.csv

phone\_brand\_device\_model.csv

sample\_submission.csv



# 03

## 文本挖掘



# 主要内容

---

- 信息检索
- 文本分类
- 关键词提取
- 文本主题模型

# 一个信息检索的例子

- 很多人都有Shakespeare's Collected Works (《莎士比亚全集》) 这本大部头的书。假定你想知道其中的哪些剧本包含 Brutus 和 Caesar 但不包含Calpurnia。
- 一种办法就是从头到尾阅读这本全集，对每部剧本都留心它是否包含Brutus 和 Caesar且同时不包含Calpurnia。这种线性扫描就是一种最简单的计算机文档检索方式。
- 很多情况下只采用上述扫描方式是远远不够的，我们需要做更多的处理：
  - 大规模文档集条件下的快速查找
  - 有时我们需要更灵活的匹配方式
  - 需要对结果进行排序
- 倒排索引的数据结构

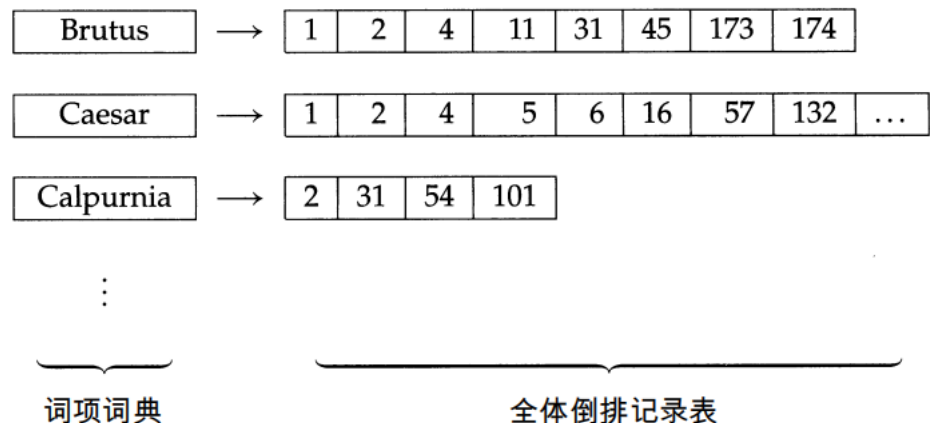


图 1-3 倒排索引的两个部分。词典部分往往放在内存中，而指针指向的每个倒排记录表则往往存放在磁盘上

# 倒排索引

## • 步骤

为获得检索速度的提升，就必须事先建立索引。建立索引的主要步骤如下。

(1) 收集需要建立索引的文档，如：

Friends, Romans, countrymen. So let it be with Caesar...

(2) 将每篇文档转换成一个个词条<sup>①</sup> ( token ) 的列表，这个过程通常称为词条化 ( tokenization )，如：

Friends Romans countrymen So...

(3) 进行语言学预处理，产生归一化的词条来作为词项，如：

Friends roman countrymen So...

4. 对所有文档按照其中出现的词项来建立倒排索引，索引中包括一部词典和一个全体倒排记录表。

# 词项集合的确定

- 词条化

输入：Friends, Romans, Countrymen, lend me your ears;  
输出：

Friends	Romans	Countrymen	lend	me	your	ears
---------	--------	------------	------	----	------	------

- 去除停用词

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

图 2-5 Reuters-RCV1 语料库中的 25 个停用词

查询词项	文档中应当匹配的项
Windows	Windows
windows	Windows, windows, window
window	window, windows

图 2-6 能够对用户期望进行有效建模的不对称扩展的例

- 词项归一化

- 是将看起来不完全一致的多个词条归纳成一个等价类，以便在它们之间进行匹配的过程

- 词干还原和词形合并

am, are, is ⇒ be  
car, cars, car's, cars' ⇒ car

# TF-IDF

首先，我们对于词项  $t$ ，根据其在文档  $d$  中的权重来计算它的得分。最简单的方式是将权重设置为  $t$  在文档中的出现次数。这种权重计算的结果称为词项频率 ( term frequency )，记为  $tf_{t,d}$ ，其中的两个下标分别对应词项和文档。

实际中，一个更常用到的因子是文档频率 ( document frequency )  $df_t$ ，它表示的是出现  $t$  的所有文档的数目。这是因为文档评分的目的是区分文档，所以最好采用基于文档粒度的统计量

由于  $df$  本身往往较大，所以通常需要将它映射到一个较小的取值范围中去。为此，假定所有文档的数目为  $N$ ，词项  $t$  的  $idf$  ( inverse document frequency, 逆文档频率 ) 的定义如下：

$$idf_t = \log \frac{N}{df_t}。 \quad (6-7)$$

# TF-IDF

## • 排序

$$Score(q, d) = \sum_{t \in q} \text{tf-idf}_{t,d} \circ$$

对于每篇文档中的每个词项，可以将其 tf 和 idf 组合在一起形成最终的权重。tf-idf 权重机制对文档  $d$  中的词项  $t$  赋予的权重如下：

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t \circ \quad (6-8)$$

换句话说， $\text{tf-idf}_{t,d}$  按照如下的方式对文档  $d$  中的词项  $t$  赋予权重：

- (1) 当  $t$  只在少数几篇文档中多次出现时，权重取值最大（此时能够对这些文档提供最强的区分能力）；
- (2) 当  $t$  在一篇文档中出现次数很少，或者在很多文档中出现，权重取值次之（此时对最后的相关度计算作用不大）；
- (3) 如果  $t$  在所有文档中都出现，那么权重取值最小。

# 文本分类

- 经典的NLP基本问题

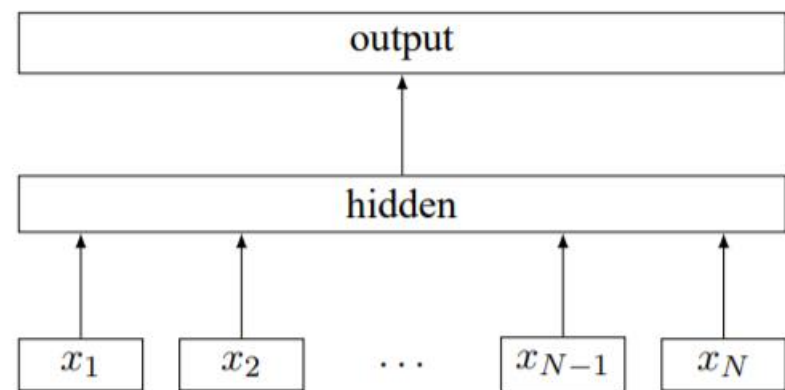
- [https://github.com/brightmart/text\\_classification](https://github.com/brightmart/text_classification)
- <https://github.com/649453932/Chinese-Text-Classification-Pytorch>
- <https://github.com/yongzhuo/Keras-TextClassification>

- 算法

- 传统算法SVM
- 深度学习相关: FastText, TextCNN, TextRNN
  - <https://github.com/facebookresearch/fastText>

- 数据

- 维基百科
  - <http://commoncrawl.org/>
- 搜狗实验室
  - [http://www.sogou.com/labs/resource/list\\_pingce.php](http://www.sogou.com/labs/resource/list_pingce.php)
- 中文文本分类数据集THUCNews
  - <http://thuctc.thunlp.org/#%E4%B8%AD%E6%96%87%E6%96%87%E6%9C%AC%E5%88%86%E7%B1%BB%E6%95%B0%E6%8D%AE%E9%9B%86THUCNews>



**Figure 1:** Model architecture of `fastText` for a sentence with  $N$  ngram features  $x_1, \dots, x_N$ . The features are embedded and averaged to form the hidden variable.

# 关键词提取

---

- 基于统计特征的关键词提取算法
  - 核心：利用文档中词语的统计信息抽取文档的关键词
  - 步骤
    - 文本经过预处理得到候选词语的集合
    - 采用特征值量化的方式从候选集合中得到关键词
    - 关键是采用什么样的特征值量化指标的方式
  - 特征值量化指标
    - 基于词权重的特征量化
      - 基于词权重的特征量化主要包括词性、词频、逆向文档频率、相对词频、词长等。
    - 基于词的文档位置的特征量化
      - 根据文章不同位置的句子对文档的重要性不同的假设来进行的。
      - 文章的前N个词、后N个词、段首、段尾、标题、引言等位置的词具有代表性，这些词作为关键词可以表达整个的主题
    - 基于词的关联信息的特征量化
      - 词的关联信息是指词与词、词与文档的关联程度信息，包括互信息、hits值、贡献度、依存度、TF-IDF值等。
- 参考：如何做好文本关键词提取？从三种算法说起，<https://www.jiqizhixin.com/articles/2018-11-14-17>



# 文本主题模型

- 算法
  - LDA, Latent Dirichlet Allocation (LDA)
- 工具
  - Gensim
  - 百度
    - <https://github.com/baidu/Familia>
- 例子
  - Gensim:  
[https://blog.csdn.net/Yellow\\_python/article/details/83097994](https://blog.csdn.net/Yellow_python/article/details/83097994)
  - Baidu
    - <https://github.com/baidu/Familia/wiki/%E4%BD%BF%E7%94%A8%E6%96%87%E6%A1%A3%E2%80%94%E2%80%94%E8%AF%AD%E4%B9%89%E8%A1%A8%E7%A4%BA>

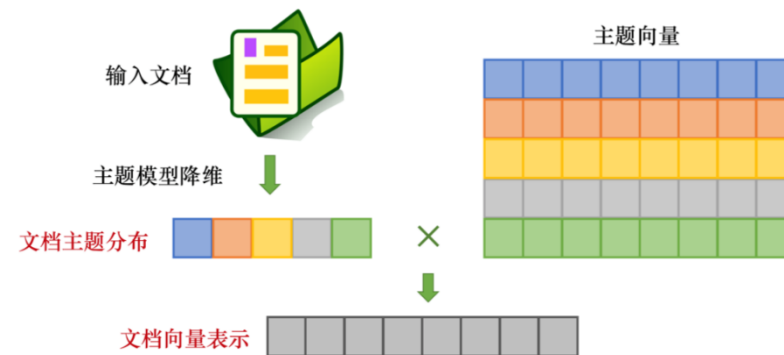


图 1: 文档的特征表示



build passing license BSD

Familia 开源项目包含文档主题推断工具、语义匹配计算工具以及基于工业级语料训练的三种主题模型: Latent Dirichlet Allocation(LDA)、SentenceLDA 和 Topical Word Embedding(TWE)。支持用户以“拿来即用”的方式进行文本分类、文本聚类、个性化推荐等多种场景的调研和应用。考虑到主题模型训练成本较高以及开源主题模型资源有限的现状,我们会陆续开放基于工业级语料训练的多个垂直领域的主题模型,以及这些模型在工业界的典型应用方式,助力主题模型技术的科研和落地。  
([English](#))

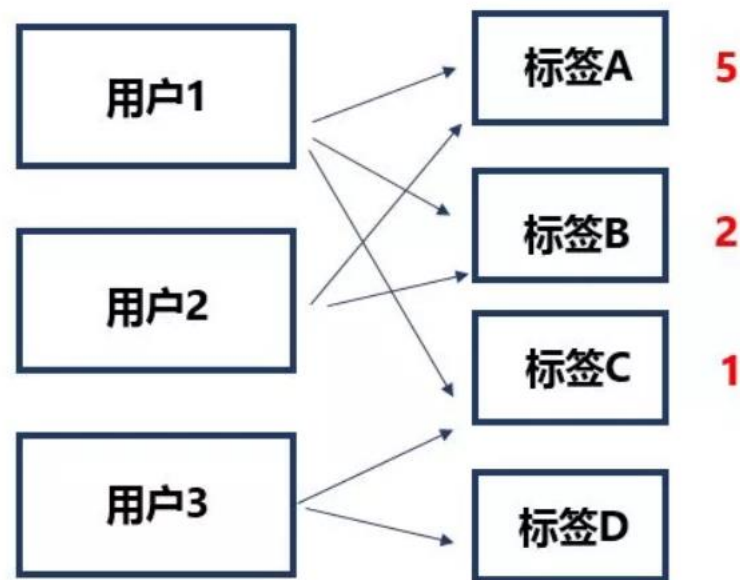
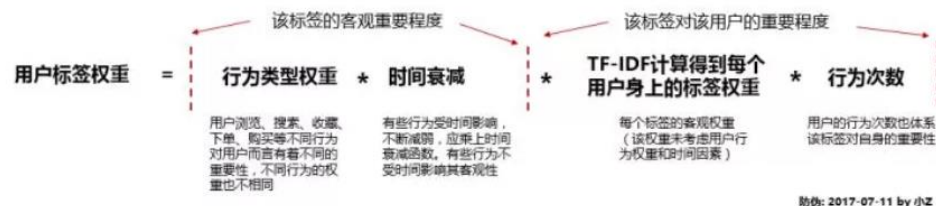


# 04

## 权重计算

# 概述

- 记录用户曝光点击历史
- 提取文章相关标签
- 曝光且点击加权
- 曝光不惦记降权
- 热门降权
- 按照时间衰减
- 归一化



- 参考
  - 用户画像之标签权重算法
  - [https://mp.weixin.qq.com/s?\\_\\_biz=MzI0OTQyNzEzMQ==&mid=2247487211&idx=1&sn=848069327f8c778e42427158f20f9b36&chksm=e990eb3fdee7622915479093a8f43f61dc8772cc681498f95dbde6960f11c5ed8f75bde29a8e&scene=21#wechat\\_redirect](https://mp.weixin.qq.com/s?__biz=MzI0OTQyNzEzMQ==&mid=2247487211&idx=1&sn=848069327f8c778e42427158f20f9b36&chksm=e990eb3fdee7622915479093a8f43f61dc8772cc681498f95dbde6960f11c5ed8f75bde29a8e&scene=21#wechat_redirect)

# 基于TF-IDF算法的权重归类

$w(P, T)$  表示一个标签T被用于标记用户P的次数

$$TF(P, T) = \frac{w(P, T)}{\sum_{T_i \in \text{该用户全部标签}} w(P, T_i)}$$

——→ 打在某用户身上某个标签的个数

——→ 该用户身上全部标签个数

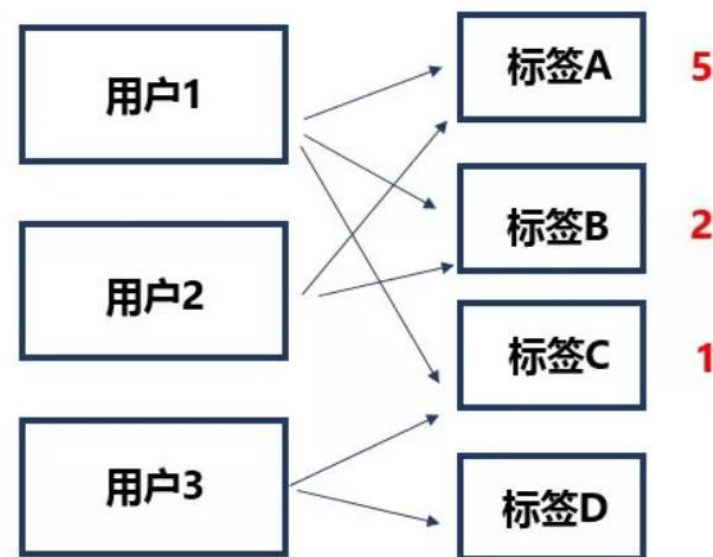
$T_i \in \text{该用户全部标签}$

$$IDF(P, T) = \frac{\sum \sum w(P_j, T_i)}{\sum_{P_i \in \text{全部用户}} w(P_i, T)}$$

——→ 全部用户的全部标签之和

——→ 所有打T标签的用户之和

$P_i \in \text{全部用户}$



# 时间衰减

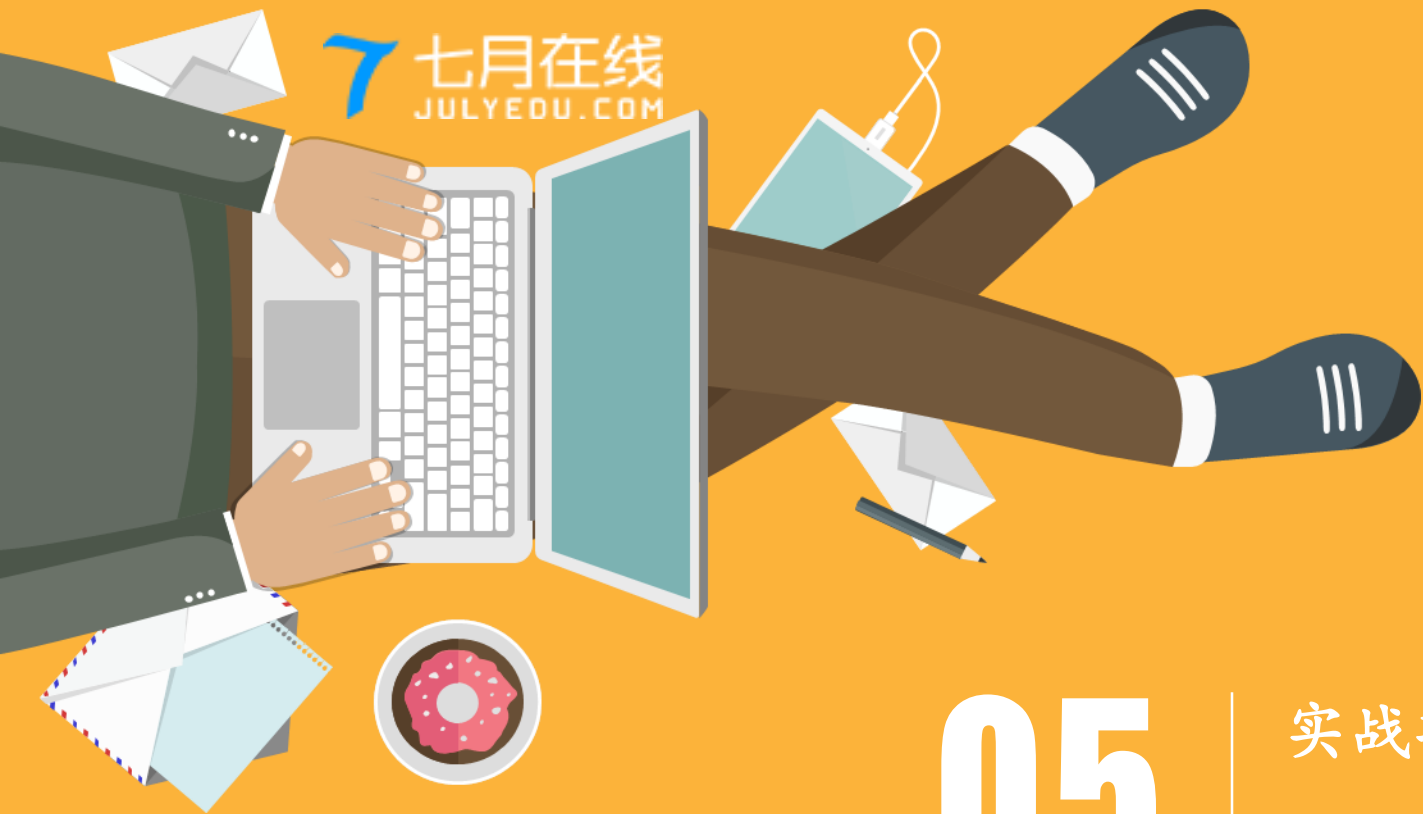
---

- 牛顿冷却定律数学模型
- $w(P, T)$  表示一个标签 $T$ 被用于标记用户 $P$ 的次数
- 不同时间的 $w(P, T)$ 可以做时间衰减
- 指定45分钟后物体温度为初始温度的0.5, 即  $0.5 = 1 \times \exp(-a \times 45)$ , 求得 $\alpha = 0.1556$ 。

# 多维度多层次

---

- 用户长期/短期行为
  - 需要分开计算
  - 短期行为多采用实时架构
- 用户精确兴趣/泛化兴趣
  - 精确兴趣通过点击行为得到
  - 泛化兴趣通过精确兴趣+近似算法扩充



# 05

实战项目：倒排索引/人口学属性预测

# 查询引擎构建

---

参考:

<http://mocilas.github.io/2015/11/18/Python-Inverted-Index-for-dummies/>

<https://github.com/matteobertozzi/blog-code/blob/master/py-inverted-index/invindex.py>

```
doc1 = """
Niners head coach Mike Singletary will let Alex Smith remain his starting
quarterback, but his vote of confidence is anything but a long-term mandate.

Smith now will work on a week-to-week basis, because Singletary has voided
his year-long lease on the job.

"I think from this point on, you have to do what's best for the football team,"
Singletary said Monday, one day after threatening to bench Smith during a
27-24 loss to the visiting Eagles.
"""
```

```
doc2 = """
The fifth edition of West Coast Green, a conference focusing on "green" home
innovations and products, rolled into San Francisco's Fort Mason last week
intent, per usual, on making our living spaces more environmentally friendly
- one used-tire house at a time.
```

```
To that end, there were presentations on topics such as water efficiency and
the burgeoning future of Net Zero-rated buildings that consume no energy and
produce no carbon emissions.
```

```
queries =
['Week',
 'Niners week',
 'West-coast Week']
```



# 构建倒排索引

---

- word\_split
- words\_cleanup
- words\_normalize
- word\_index
- inverted\_index
- inverted\_index\_add
- search



微信扫一扫关注我们

# THANKS

---

<https://www.julyedu.com/>

---