

# Douyin广告分类

Qian Xu

Feb 2019

# 1 项目背景

# 目录

- 项目背景
- 处理方法
  - 数据预处理
  - 建模分析
  - 结果比较
- 全量广告分类
- 参考资料

 抖音短视频

下载抖音 看快乐大本营独家花絮



# 1 项目背景

元旦前后，有大量广告投放在抖音，尝试通过数据建模的方式对抖音广告进行分类统计，提高分类效率和准确率。

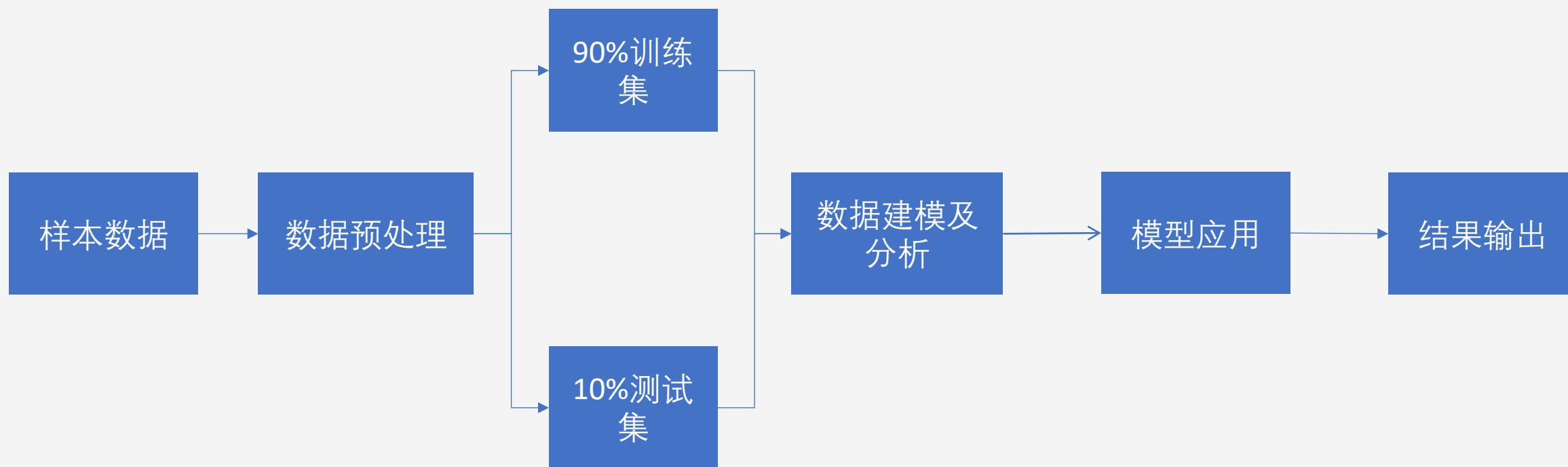
时间：2018.01.01 ~ 2018.12.20

总体数据规模：约35万条

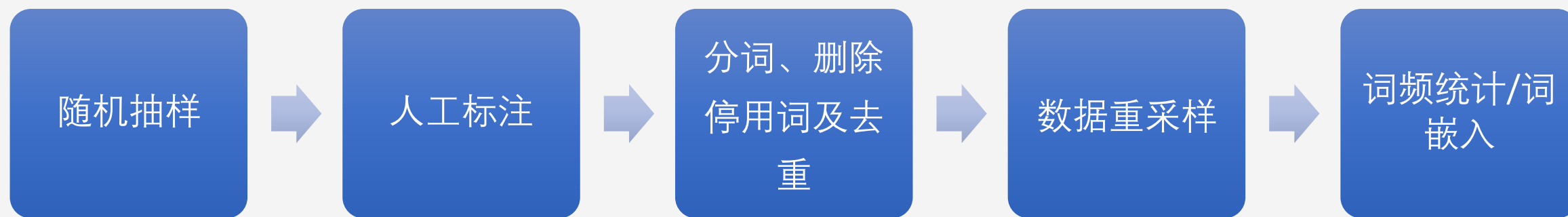
实验数据：2万条

## 2 处理方法

## 2 整体流程



## 2 数据预处理



- 随机选择一批广告数据并人工标注类别（2万条）

护肤美容,透蜜旗舰店-素颜霜,12.12一抹懒人素颜女神仅59.9。淘宝搜:透蜜旗舰店  
护肤美容,透蜜旗舰店-素颜霜,急救懒人霜,打造素颜女神。淘宝搜:透蜜旗舰店  
护肤美容,透蜜旗舰店-素颜霜,透蜜素颜霜,一抹懒人素颜女神,立即抢购!透蜜旗舰店  
护肤美容,透蜜旗舰店-素颜霜,透蜜素颜霜,一抹懒人素颜女神!赶紧加购吧  
护肤美容,透蜜旗舰店-素颜霜,现在最火的素颜霜,到底是什么来头?淘宝:透蜜旗舰店  
护肤美容,透蜜旗舰店-素颜霜,最近超火的素颜霜真的好用吗?淘宝搜:透蜜旗舰店  
护肤美容,透真旗舰店-素颜霜,我为歌狂,身在本地的你,一起来参加年终歌会  
护肤美容,丸碧-诚招创业合伙人,免费加盟的机会还不看看?限本省人报名  
护肤美容,丸碧-诚招创业合伙人,对不起!我们来晚了,本月才招本地城市合伙人  
护肤美容,丸碧-诚招创业合伙人,闺蜜多年不见变化这么大!原来加盟了丸碧美妆  
护肤美容,丸碧-诚招创业合伙人,免费加盟丸碧美妆,0经验全程培训,还享特权福利  
护肤美容,丸碧-诚招创业合伙人,抓紧!丸碧美妆本月再次招城市合伙人  
护肤美容,丸碧-诚招创业合伙人,在家没事做?不如免费加盟美妆好项目,还享特权福利  
护肤美容,丸碧-诚招创业合伙人,化妆品随便用!免费加盟丸碧美妆,享受特权福利  
游戏,王国纪元-官方推荐,据说,智商120以上的大神,才能玩好这游戏的排兵布阵  
游戏,王国纪元-官方推荐,这款策略游戏有什么魔力?竟然让男友天天玩到凌晨3点  
游戏,王国纪元-官方推荐,玩了这个游戏,短期内不会换游戏了  
游戏,王国纪元-官方推荐,真正能带兵打仗的策略手游,只烧脑不烧钱,仅限IOS  
游戏,王国纪元-官方推荐,本地小伙半路埋伏老外,团灭敌军8000,发财了



- 分词、去停用词和重复数据
  - 利用python中文分词包jieba对广告描述语进行分词
  - 下载stopwords list , 删除分词中的标点符号以及的、地、得、呀等无意义词
  - 去重

```
out_feeds_df.iloc[11815:11817]['desp']
```

```
11815    欧美爆款马丁靴，好穿不臭脚！支持货到付款！开箱试穿
```

```
11816    欧美爆款马丁靴！好穿不臭脚！支持货到付款！开箱试穿
```

```
Name: desp, dtype: object
```

```
seg_sentence(out_feeds_df.iloc[11815]['desp'], stopwords_list)
```

```
'欧美 爆款 马丁 靴 穿 臭脚 支持 货到付款 开箱 试穿 '
```

```
seg_sentence(out_feeds_df.iloc[11816]['desp'], stopwords_list)
```

```
'欧美 爆款 马丁 靴 穿 臭脚 支持 货到付款 开箱 试穿 '
```

## 数据预处理

- 问题：由于某些类别数据量偏少，可能导致该类数据分类误差较大。
- 解决方式：数据重采样。对于本例，将第七类旅游出行重复一倍，第八类日用百货重复两倍。

	class	account	desp
0	家居家装	1578	1578
1	房地产	590	590
2	护肤美容	1009	1009
3	摄影	454	454
4	教育培训	1764	1764
5	文化娱乐	630	630
6	新闻	670	670
7	旅游出行	278	278
8	日用百货	189	189
9	服饰鞋包	367	367
10	汽车	783	783
11	游戏	986	986
12	珠宝腕表	935	935
13	生活服务	860	860
14	社交	912	912
15	综合电商	2169	2169
16	视频音频	624	624
17	运动健身	1207	1207
18	金融借贷	592	592
19	餐饮食品	418	418

	class	account	desp
0	家居家装	1578	1578
1	房地产	590	590
2	护肤美容	1009	1009
3	摄影	454	454
4	教育培训	1764	1764
5	文化娱乐	630	630
6	新闻	670	670
7	旅游出行	556	556
8	日用百货	567	567
9	服饰鞋包	367	367
10	汽车	783	783
11	游戏	986	986
12	珠宝腕表	935	935
13	生活服务	860	860
14	社交	912	912
15	综合电商	2169	2169
16	视频音频	624	624
17	运动健身	1207	1207
18	金融借贷	592	592
19	餐饮食品	418	418

## 数据预处理

- 生成语料库：尽量使用全部语料，包括训练数据和待预测数据。
  - 语料一：仅使用广告描述词
  - 语料二：使用广告描述词和广告主名称

脸上 满是 痘痘 难看 痘 博士 21.5 面部 祛痘  
福利 姿势 挤 痘 预约 21.5 专业 面部 祛痘  
脸上 痘 苦恼 预约 痘 博士 21.5 面部 祛痘 套餐  
告诉 秘密 领券后 买 衣服 不到 50 太值  
领券后 下单 每套 衣服 不到 50 元  
同事 新 衣服 买 领完券 一套 不到 50 元  
告诉 秘密 领券后 买 衣服 50 元 太值  
同事 衣服 买 一套 50 元 超 划算  
秋季 福利 买 一套 衣服 50 元 包邮  
同事 新 衣服 买 领完券 一套 50 元  
脸上 痘痘 毛孔 清洁 预约 21.5 面部 祛痘 套餐

语料一：仅使用广告描述词

痘 博士 脸上 满是 痘痘 难看 痘 博士 21.5 面部 祛痘  
痘 博士 福利 姿势 挤 痘 预约 21.5 专业 面部 祛痘  
痘 博士 脸上 痘 苦恼 预约 痘 博士 21.5 面部 祛痘 套餐  
痘 博士 告诉 秘密 领券后 买 衣服 不到 50 太值  
痘 博士 领券后 下单 每套 衣服 不到 50 元  
痘 博士 同事 新 衣服 买 领完券 一套 不到 50 元  
痘 博士 告诉 秘密 领券后 买 衣服 50 元 太值  
痘 博士 同事 衣服 买 一套 50 元 超 划算  
痘 博士 秋季 福利 买 一套 衣服 50 元 包邮  
痘 博士 同事 新 衣服 买 领完券 一套 50 元  
痘 博士 脸上 痘痘 毛孔 清洁 预约 21.5 面部 祛痘 套餐

语料二：使用广告描述和广告主名称

## 2 数据预处理----TF-IDF

TF-IDF是一种统计方法，用以评估一字/词对一个文件集或者一个语料库中其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

- TF ( Term Frequency ) : 词频

$$tf_{i,j} = n_{i,j} / \sum_k n_{k,j}$$

$n_{i,j}$  是词  $t_i$  在文件  $d_j$  中出现的次数，分母是文件  $d_j$  中所有字词出现的次数

- IDF ( Inverse document frequency ) : 逆向文件频率

$$idf_i = \log \left| D / \left| \{ j : t_i \in d_j \} \right| \right|$$

分子  $|D|$  表示语料库中的文件总数，分母表示含词语  $t_i$  的文件数目，若  $t_i$  不存在，则加1。

- TF-IDF = TF × IDF

## 2 数据预处理----TF-IDF

- Corpus = [ “我 来到 北京 清华大学” , “ 他 来到了 网易 杭研 大厦” , “ 小明 硕士 毕业 与 中国 科学院” , “ 我 爱 北京 天安门” ]

关键词	中国	北京	大厦	天安门	小明	来到	杭研	毕业	清华大学	硕士	科学院	网易
频次	0	1	0	0	0	1	0	0	1	0	0	0
	0	0	1	0	0	1	1	0	0	0	0	1
	1	0	0	0	1	0	0	1	0	1	1	0
	0	1	0	1	0	0	0	0	0	0	0	0
TF-IDF 矩阵	0	0.526	0	0	0	0.526	0	0	0.668	0	0	0
	0	0	0.525	0	0	0.414	0.525	0	0	0	0	0.525
	0.447	0	0	0	0.447	0	0	0.447	0	0.447	0.447	0
	0	0.619	0	0.785	0	0	0	0	0	0	0	0

## 2 数据预处理----TF-IDF

- 抖音广告TF-IDF矩阵：

```
[ [0. 0. 0. ... 0. 0. 0.]  
  [0. 0. 0. ... 0. 0. 0.]  
  [0. 0. 0. ... 0. 0. 0.]  
  ...  
  [0. 0. 0. ... 0. 0. 0.]  
  [0. 0. 0. ... 0. 0. 0.]  
  [0. 0. 0. ... 0. 0. 0.] ]  
65971 45060
```

- TF-IDF优点：
  - 简单快速，容易理解
- TF-IDF缺点：
  - 仅用词频衡量某个词的重要性不够全面
  - 无法体现词的位置信息和在上下文里的重要性



## 2 数据预处理----word2vec

- 定义：建立在神经网络概率语言模型（Neural Probabilistic Language Model）基础上，某些用来产生词嵌入（word embedding）的相关模型或工具。
- 基础知识：
  - 概率语言模型： $P(s) = P(w_1, w_2, \dots, w_n) = P(w_1) * P(w_2 | w_1) * \dots * P(w_n | w_1, w_2, \dots, w_{n-1}) = \sum P(w_i | content_i)$
  - N-gram模型：为了降低计算复杂度，在计算某项概率时仅考虑该词前面的N-1个词，（N=1,2,...,5）。
  - 词向量：

One-hot Representation：向量的维度是词表的大小，比如有10万个词，该词向量维度为10万。

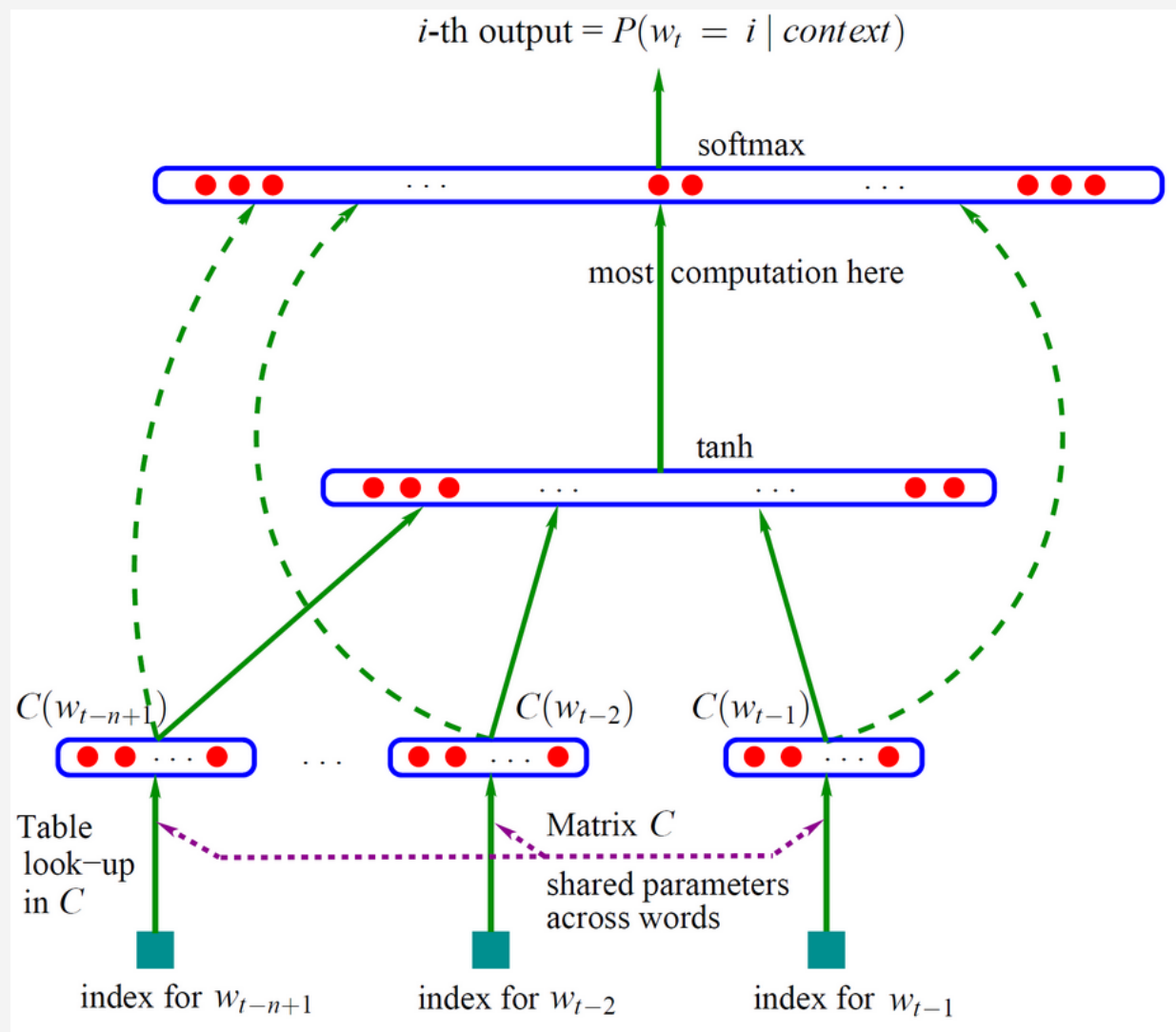
```
v('足球') = [0 1 0 0 0 0 0 0 .....]  
v('篮球') = [0 0 0 0 0 0 1 0 .....]
```

Distributed Representation：向量的维度是某个具体的值，如50

```
v('足球') = [0.26 0.49 -0.54 -0.08 0.16 0.76 0.33 .....]  
v('篮球') = [0.31 0.54 -0.48 -0.01 0.28 0.94 0.38 .....]
```

## 2 数据预处理----word2vec

神经网络概率语言模型：NNLM

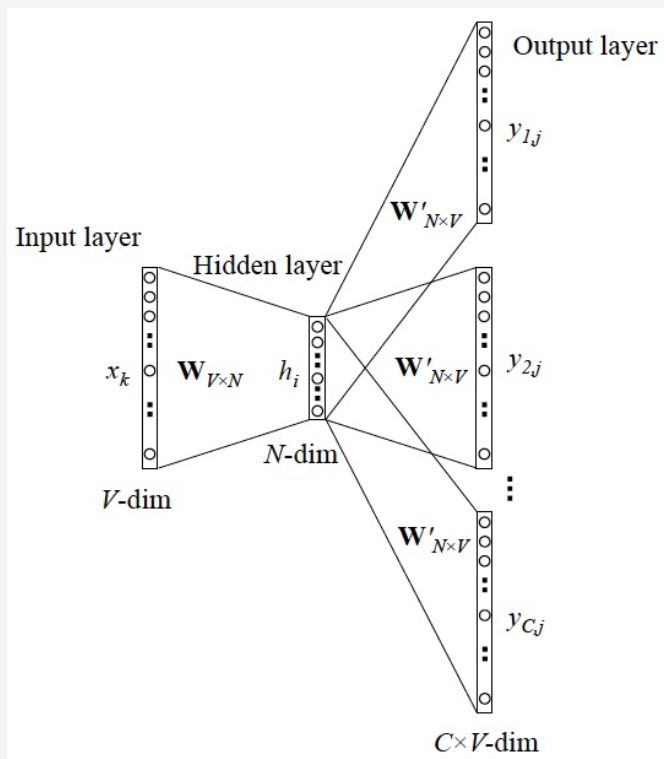




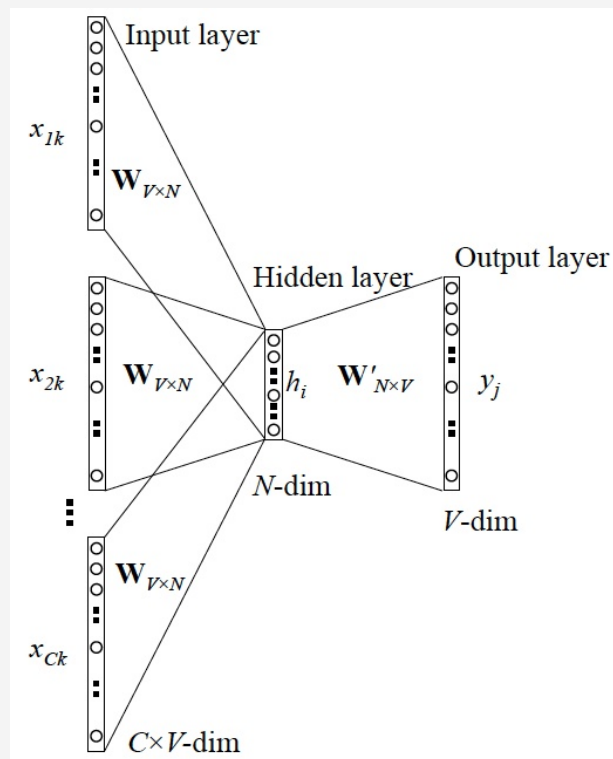
# 数据预处理----word2vec

代表模型：

- Skip-gram：已知中心词，预测该词可能的上下文。（应用更广）
- CBOW：已知上下文，预测可能的中心词。



Skip-gram



CBOW

## 2 数据预处理----word2vec

- 例1：找出语料中和南京最接近的3个词？

```
model.wv.most_similar('南京', topn=3)
```

```
2019-01-14 15:59:31,237:INFO:precomputi
/Library/Frameworks/Python.framework/Ve
Conversion of the second argument of is
treated as `np.int64 == np.dtype(int).t
    if np.issubdtype(vec.dtype, np.int):
```

```
[('婚纱', 0.9507881999015808),
 ('尊荣', 0.9396384954452515),
 ('米兰', 0.9387040138244629)]
```

- 例2：计算装修和南京两个词的相似度？

```
model.wv.similarity('南京', '装修')
```

```
/Library/Frameworks/Python.framework
Conversion of the second argument c
treated as `np.int64 == np.dtype(ir
    if np.issubdtype(vec.dtype, np.ir
```

```
0.44735348
```

## 2 模型分类----逻辑回归

- 定义：一种广义的线性回归，与多重线性回归有相同之处，区别在于因变量不同。逻辑回归是常用的二分类机器学习算法，但也可用于多分类问题。

- 线性回归函数： $y = \beta_0 + \beta \cdot x$

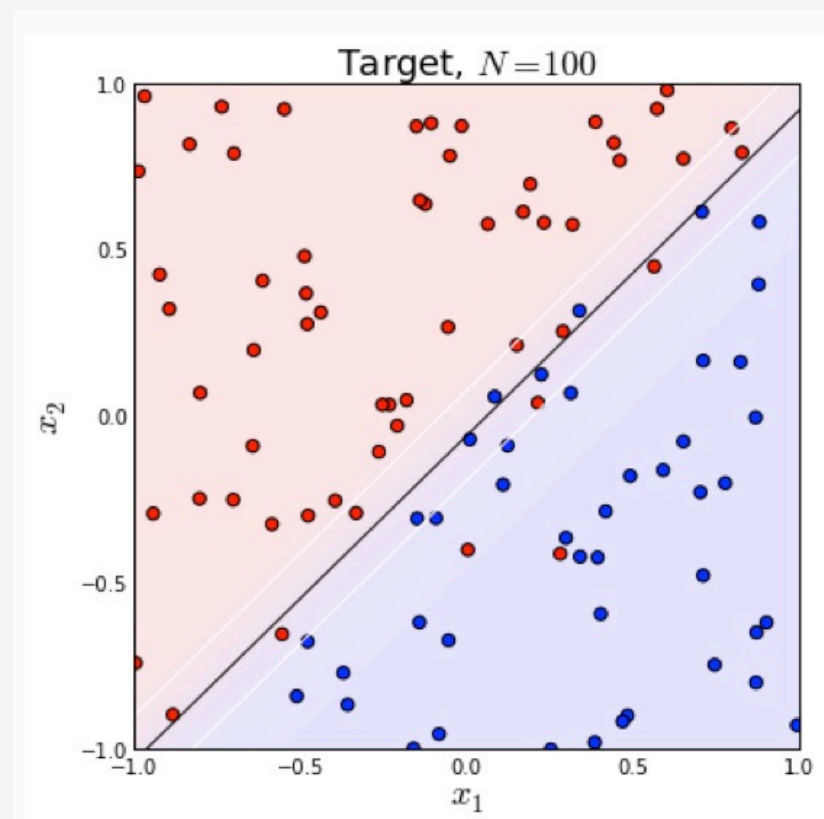
- Sigmoid函数： $g(x) = \frac{1}{1 + e^{-x}}$

- 逻辑回归模型： $\log \frac{p(x)}{1 - p(x)} = \beta_0 + x \cdot \beta$

$$y = \begin{cases} 1, p \geq 0.5 \\ 0, p < 0.5 \end{cases} \quad p(x; b, w) = \frac{e^{\beta_0 + x \cdot \beta}}{1 + e^{\beta_0 + x \cdot \beta}} = \frac{1}{1 + e^{-(\beta_0 + x \cdot \beta)}}$$

- 似然函数： $L(\beta_0, \beta) = \prod_{i=1}^n p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$

$$l(\beta_0, \beta) = \sum_{i=1}^n (y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)))$$

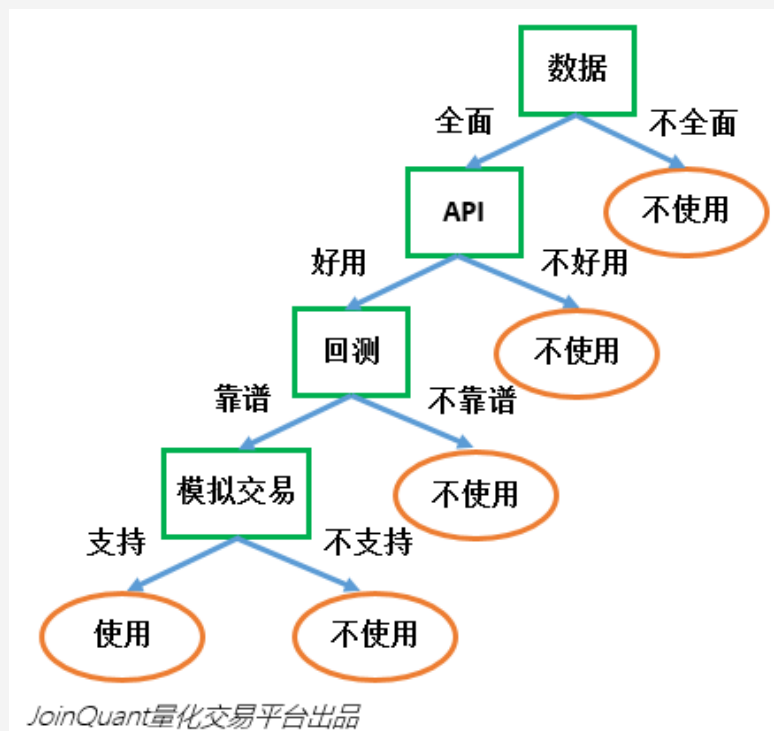


## 2 模型分类----逻辑回归

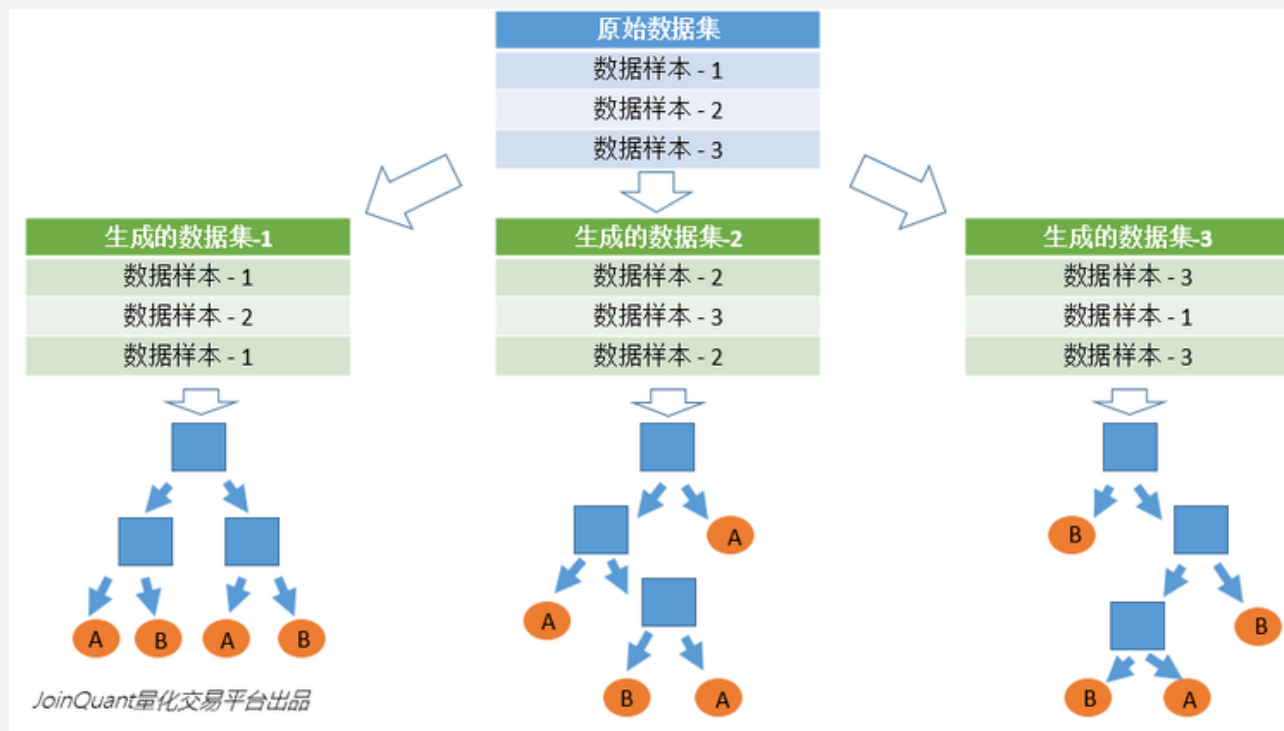
- 优点：
  - 适合分类场景的分析
  - 计算代价不高，容易理解实现，可以用较少资源处理大型数据
  - 抗噪声干扰能力强，可以通过L1和L2正则化的方法选择特征避免过度拟合。
  - 容易解释，应用简单
- 缺点：
  - 准确率一般。因为形式非常简单（类似线性模型），很难拟合数据的真实分布。
  - 很难处理数据不平衡的问题。
  - 处理非线性数据比较麻烦。

## 2 模型分类----随机森林

- 随机森林是一个包含多个独立决策树的分类器，并且其输出的类别是有个别树输出的类别的众数而定。



决策树



随机森林

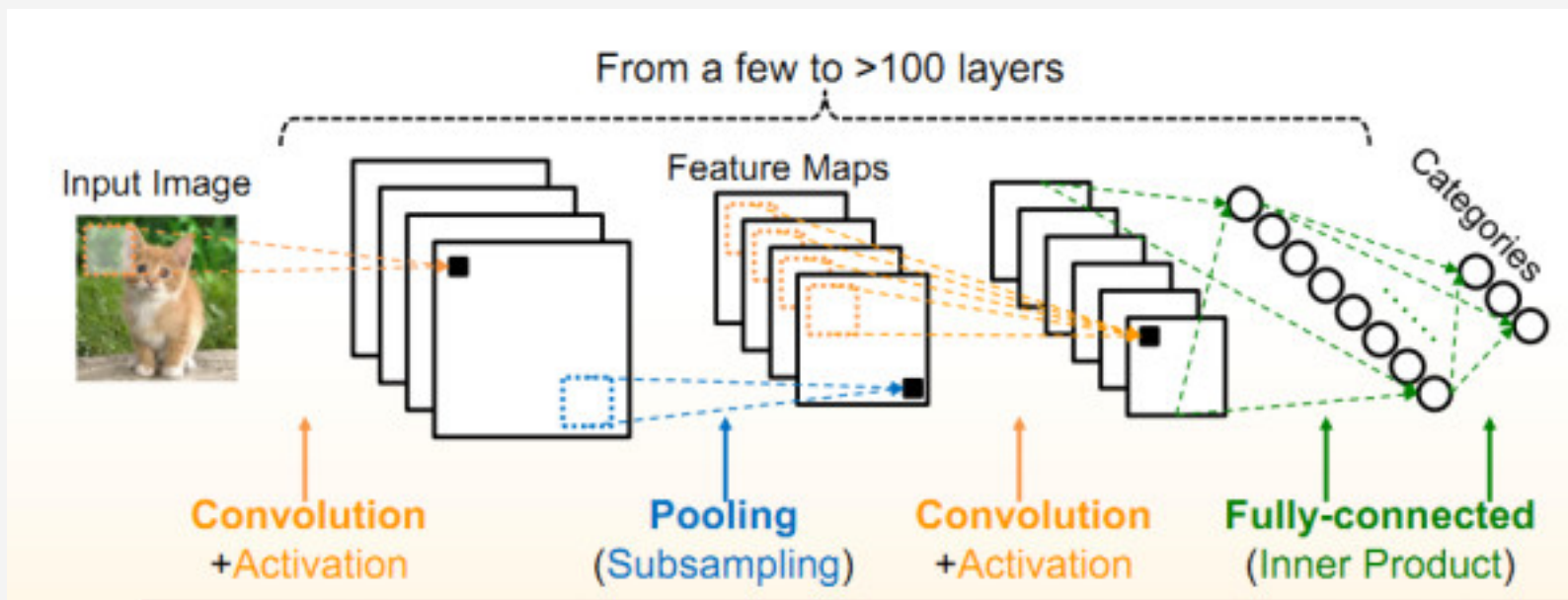
Ps : 1. 有放回抽样，各子集中有重复数据 2. 各子决策树独立，特征随机选取

## 2 模型分类----随机森林

- 优点：
  - 能够处理高维度（feature很多）数据，由于特征子集随机选择，所以不用做特征选择
  - 训练时树和树之间相互独立，训练速度快，容易做成并行化方法
  - 对大规模数据集和存在大量不相关特征的数据很有用
  - 训练完成后，能够给出特征重要性
  - 对generalization error使用的是无偏估计，模型泛化能力强
  - 对缺失数据不敏感，即使有很大部分的特征遗失，仍可以维持准确度
  - 对不平衡的数据集来说，可以平衡误差
- 缺点：
  - 在某些噪音较大的分类或回归问题上可能会过拟合
  - 对于有不同取值的特征数据，级别划分较多的特征对随机森林产生更大影响，导致在该特征上产生的权值不可信

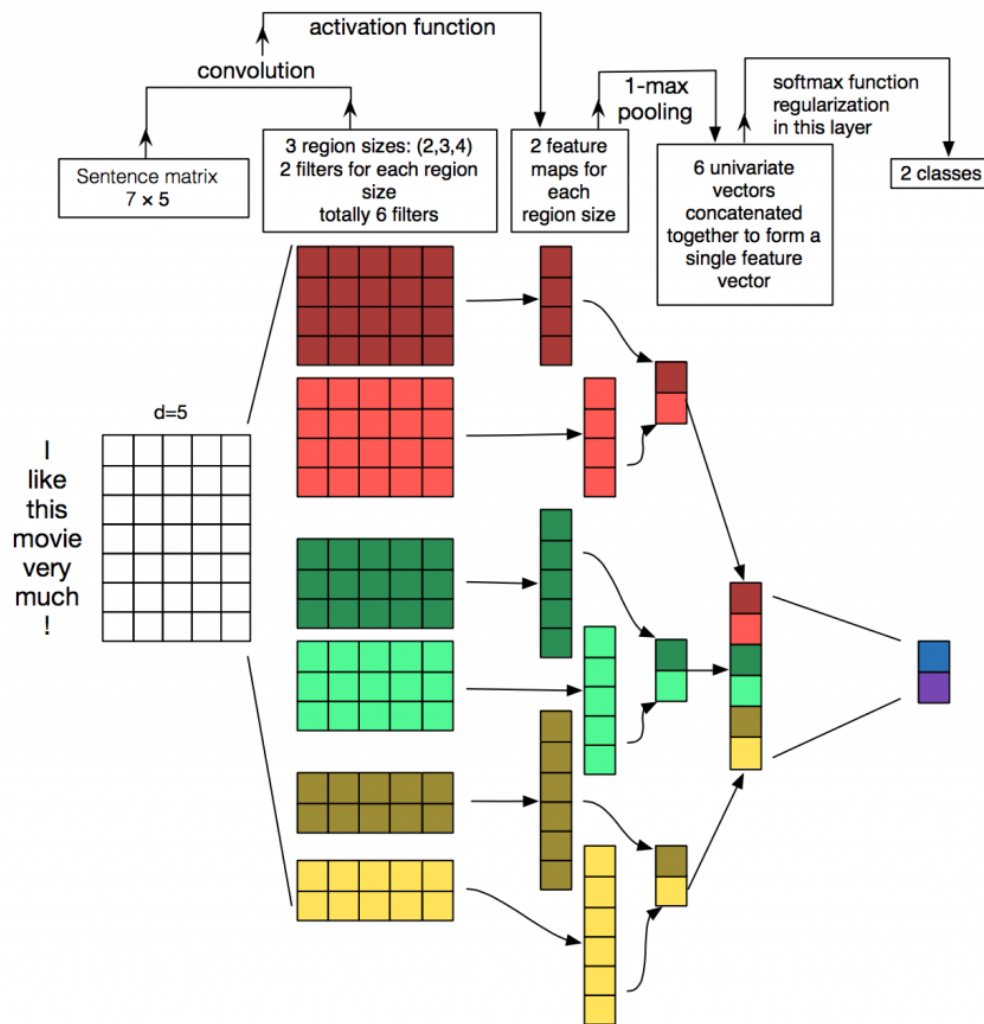
## 2 模型分类----CNN

- CNN（卷积神经网络）是一种前馈神经网络，它由一个或多个卷积层和顶端的全连通层组成，同时也包括关联权重和池化层（pooling layer）。与其他深度学习结构相比，CNN在图像和语音识别方面能给出更好的结果。



## 2 模型分类----文本分类CNN

- 在深度学习中，递归神经网络（RNN）是主流NLP处理算法，RNN擅长时序特征捕获，所以不受文本长度限制，但CNN计算速度快，对于广告语之类的短句，CNN更有优势。





## 2 结果分析

特征	模型	accuracy
TF-IDF (desp)	逻辑回归	0.8659
TF-IDF (desp)	随机森林	0.8834
TF-IDF (account + desp)	逻辑回归	0.9915
TF-IDF (account + desp)	随机森林	0.9943
Word2vec (account + desp)	CNN	0.9615

# 3 全量广告分类

## 全量广告分类

- 对全量account+desp生成语料库，并计算TF-IDF矩阵，再使用随机森林模型对未分类广告进行分类。
- 问题：由于一个广告账号对应不同的广告描述，同一广告账号可能被分到不同的类别中。

教育培训,英语流利说,用着iPhone手机还不会说英语

教育培训,英语流利说,还不敢跟外国小姐姐打招呼? 来这里让你英语沟通无压力

教育培训,英语流利说,买iPhone不学英语太亏了, 每天几分钟, 英语流利说

教育培训,英语流利说,在本地, 无论是半包还是全包, 超过这个价就亏了, 英语流利说

教育培训,英语流利说,这样的全屋定制家具竟然不到两万, 还护肤美容,爆品汇电子商务,别让大黄牙拉低了你的颜值和身份! 99元变成大白牙

教育培训,英语流利说,本区全屋定制前100用户免费量尺+设计+护肤美容,爆品汇电子商务,还你一口洁白无瑕的牙齿! 焕白牙齿就是这么简单

教育培训,英语流利说,每天5分钟, 坚持1个月, 开口说英语不家居家装,北京源色创意,餐厅门帘、日式门帘、服装店门帘-壹源色专业定制

教育培训,英语流利说,传统英语班没有时间去学? 那就看看小家居家装,北京源色创意,我在本区, 你在哪? 方便和我聊聊天吗

教育培训,英语流利说,终于找到了学不好英语的原因! 原来不家居家装,北京源色创意,附近的美女都在这交友, 你还等啥

家居家装,北京源色创意,同事们天天划手机, 原来是在探探上找到了朋友

家居家装,北京源色创意,无聊了就来探探, 和本区附近的人聊会天

家居家装,北京源色创意,一个人无聊, 用探探来找本地附近的人聊聊天

家居家装,北京源色创意,本地小姐姐单身想交友, 不闲聊, 下载探探直接配对

家居家装,北京源色创意,家里催婚想找个本地本地的, 看上的下载探探找我

家居家装,北京源色创意,我在本区, 会照顾人, 想找人聊天

家居家装,北京源色创意,偷偷下载, 适合没人时候玩的聊天app

家居家装,北京源色创意,单身一人, 寻找本区附近的陪我聊天的人

家居家装,北京源色创意,在本区附近, 无聊能和我聊天么

护肤美容,冰希黎流沙金香水,香水控! 分享一款当今热门和小众的香水推荐

护肤美容,冰希黎流沙金香水,香水攻略春夏出行香水该怎样选

护肤美容,冰希黎流沙金香水,闻香识女人, 这一瓶香水让你更加优雅迷人

## 3

- 分类结果优化：将账号分入类别众数中；如果有不止一个众数，则比较分类概率值，将账号分入较大值所在类别。

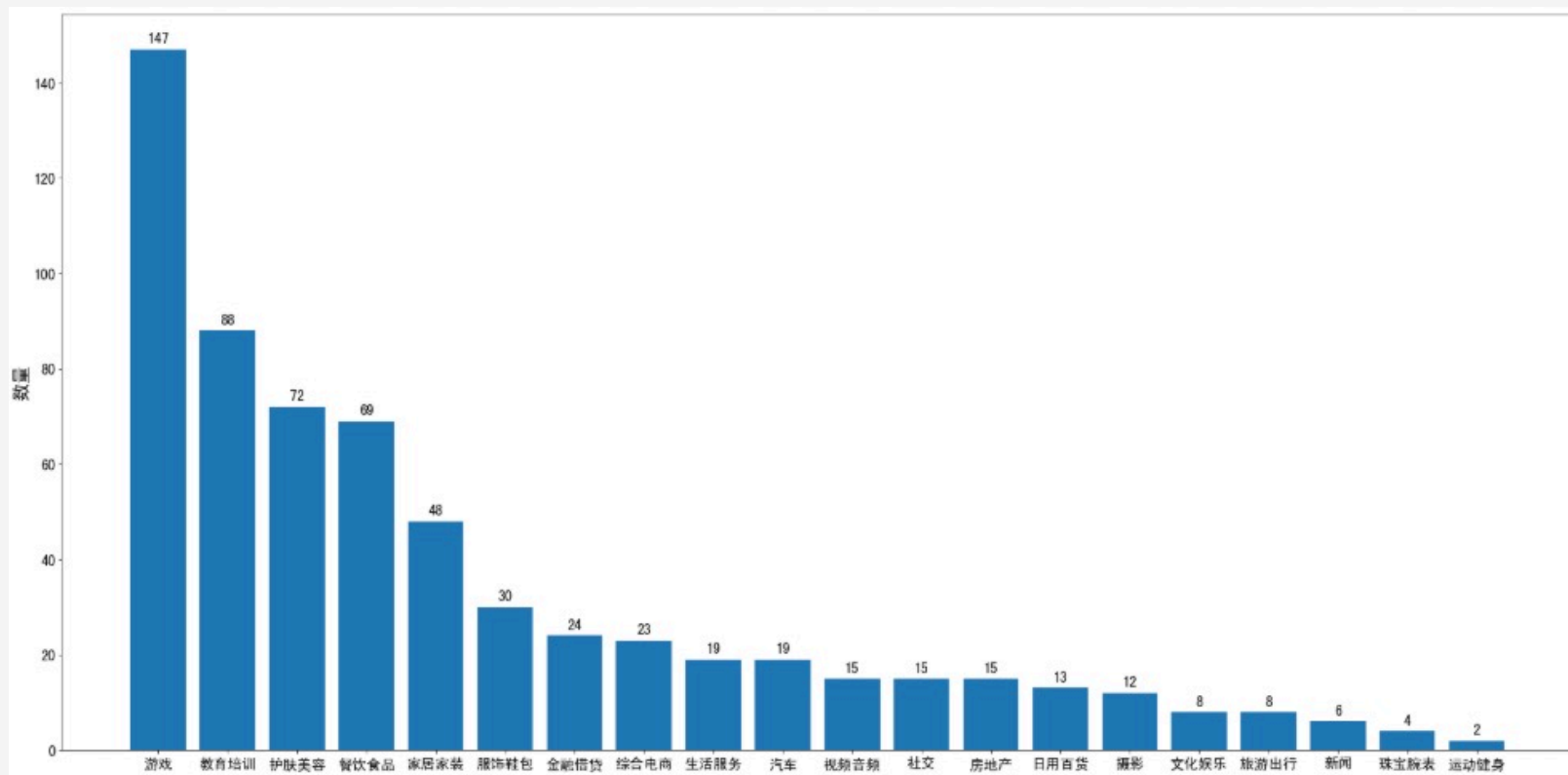
车置宝-高价收车 [ '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车',  
'汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车',  
'汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车',  
'汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '汽车', '珠宝腕表']

```
pred_tmp['智数互动-小田豆浆']
```


```
[['护肤美容',
  array([[0.02, 0., 0.56, 0., 0.04, 0., 0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0.005, 0.005, 0.005, 0., 0., 0., 0., 0., 0.],
        [0.005, 0.36]])],
 ['餐饮食品',
  array([[0.015, 0., 0.035, 0., 0.01, 0., 0., 0.01, 0., 0., 0., 0., 0.],
        [0., 0.005, 0., 0., 0., 0.005, 0.005, 0.005, 0., 0., 0., 0., 0.],
        [0., 0.91]])])]
```

### 3 方法---- word2vec+CNN

- 在工业界中，如果数据量不是特别大，传统机器学习模型已经够用。深度学习的优势在大数据中才会有明显体现。
- 对于有特点的数据，建议自己生成语料库进行训练和分析。
- 在训练阶段也尽量使用全量语料库，而不仅限于用训练样本生成语料库。



# 4 参考资料



<https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf>  
<https://tech.meituan.com/2015/05/08/intro-to-logistic-regression.html>  
<https://www.cnblogs.com/ModifyRong/p/7739955.html>  
<https://www.jiqizhixin.com/articles/2017-07-31-3>  
<https://www.joinquant.com/post/1571?f=study&m=math>  
<http://www.tensorflownews.com/2017/11/04/text-classification-with-cnn-and-rnn/>  
<https://www.aclweb.org/anthology/D14-1181>  
<http://www.pengjingtian.com/2016/09/17/nnlm/>  
<https://zhuanlan.zhihu.com/p/39579464>  
<http://www.hankcs.com/nlp/word2vec.html>  
<https://zhuanlan.zhihu.com/p/39579464>  
<http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>



# Thanks