# 基本资料（自我介绍）

您好，我叫刘鹏，今年28岁。

# 教育背景

## 武汉大学

### 数学基地班

主修课程：数学分析、高等代数与解析几何、C语言程序设计、抽象代数、数值分析、数学模型、常微分方程、实变函数、概率论与数理统计、拓扑学、微分几何、广义函数与偏微分方程、复变函数、泛函分析

## 中国科学技术大学

### 控制理论与工程

主修课程：矩阵代数、实变与泛函、随机过程理论、离散数学、计算数论、线性系统理论、非线性控制系统、鲁棒控制、机器学习

* 硕士期间发表**EI收录**会议论文1篇，和**SCI期刊**论文1篇，均为PID自动控制方向。

## 中国科学院系统科学研究院

### 复杂网络

#### 会议论文

链接：[Global Pinning Synchronization with PI Controller in General Complex Directed Networks | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/8483246)

全文：[[金山文档] CCC2018-LIUPENG-终稿.pdf](https://kdocs.cn/l/ceHaoDrwAytE)

#### 期刊论文

链接：[Global synchronization under PI/PD controllers in general complex networks with time-delay - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0925231219309737)

全文：[[金山文档] IEEE Tans Manuscript.pdf](https://kdocs.cn/l/ctQqesNcx1qM)

#### 毕业论文

[[金山文档] 毕业论文.pdf](https://kdocs.cn/l/cubq4z6VkuMw)

[[金山文档] 毕业答辩.pdf](https://kdocs.cn/l/cg7NyF5UMvOV)

### 机器学习

吴恩达机器学习笔记：特征工程决定了模型上限，之后用算法不断地去逼近这个上限。

<https://github.com/NLP-LOVE/ML-NLP>

<https://zhuanlan.zhihu.com/p/74198735>

#### 模型选择

##### 线性回归LR

##### 逻辑回归LR

##### 决策树

##### 随机森林

##### GDBT

##### XGBoost

##### LightGBM

#### 特征工程

当数据维数比较大时，就需要进行降维，特征选择是降维的一种主要方式，又包括3种方法：Filter、Wrapper和Embedded。

##### 过滤法Filter

过滤器方法，这种方法首先选定特征，再来进行学习。根据每一个属性的一些指标（如方差等），来确定这个属性的重要程度，然后对所有属性按照重要程度排序，从高到低的选择属性。选定了属性以后，再来进行训练。比如Fisher Score、Laplacian Score等。这种方法其实不大好，因为决定特征选择效果的不是单个的属性，而是属性的集合，比如属性A、B、C，单个来看效果不好，但是它们组合起来效果有可能不错。

##### 包裹法Wrapper

包裹器方法，这种方法把选定的特征集用分类器进行训练，用训练效果（如准确率等）来作为特征集的评价。比如将启发式搜索、GA等。这种方法和分类器相结合，比较直观，和Filter相比也更加合理。缺点是计算开销较大。

基本思想：基于hold-out方法，对于每一个待选的特征子集，都在训练集上训练一遍模型，然后在测试集上根据误差大小选择出特征子集。需要先选定特定算法，通常选用普遍效果较好的算法， 例如Random Forest， SVM， kNN等等。

西瓜书上说包装法应该欲训练什么算法，就选择该算法进行评估

随着学习器（评估器）的改变，最佳特征组合可能会改变

贪婪搜索算法（greedy search）是局部最优算法。与之对应的是穷举算法 (exhaustive search)，穷举算法是遍历所有可能的组合达到全局最优级，但是计算复杂度是2^n，一般是不太实际的算法。

###### 前向搜索

前向搜索说白了就是，每次增量地从剩余未选中的特征选出一个加入特征集中，待达到阈值或者 n 时，从所有的 F 中选出错误率最小的。过程如下：

初始化特征集 F 为空。

扫描 i 从 1 到 n 如果第 i 个特征不在 F 中，那么特征 i 和F 放在一起作为 F\_i (即 F\_i=F\cup{i} )。 在只使用 F\_i 中特征的情况下，利用交叉验证来得到 F\_i 的错误率。

从上步中得到的 n 个 F\_i 中选出错误率最小的 F\_i ,更新 F 为 F\_i 。

如果 F 中的特征数达到了 n 或者预定的阈值（如果有的话）， 那么输出整个搜索过程中最好的 ；若没达到，则转到 2，继续扫描。

###### 后向搜索

既然有增量加，那么也会有增量减，后者称为后向搜索。先将 F 设置为 {1,2,...,n} ，然后每次删除一个特征，并评价，直到达到阈值或者为空，然后选择最佳的 F 。

这两种算法都可以工作，但是计算复杂度比较大。时间复杂度为：[公式]

###### 递归特征消除法

递归消除特征法使用一个基模型来进行多轮训练，每轮训练后通过学习器返回的 coef\_ 或者feature\_importances\_ 消除若干权重较低的特征，再基于新的特征集进行下一轮训练。

使用feature\_selection库的RFE类来选择特征的代码如下：

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import LogisticRegression

#递归特征消除法，返回特征选择后的数据

#参数estimator为基模型

#参数n\_features\_to\_select为选择的特征个数

RFE(estimator=LogisticRegression(), n\_features\_to\_select=2).fit\_transform(iris.data, iris.target)

##### 嵌入法Embedded

嵌入式方法，即把特征选择的过程作为学习过程的一部分，在学习的过程中进行特征选择，最典型的如决策树算法。

###### 基于惩罚项的特征选择法

通过L1正则项来选择特征：L1正则方法具有稀疏解的特性，因此天然具备特征选择的特性。

from sklearn.feature\_selection import SelectFromModel

from sklearn.linear\_model import LogisticRegression

#带L1惩罚项的逻辑回归作为基模型的特征选择

SelectFromModel(LogisticRegression(penalty="l1", C=0.1)).fit\_transform(iris.data, iris.target)

要注意，L1没有选到的特征不代表不重要，原因是两个具有高相关性的特征可能只保留了一个，如果要确定哪个特征重要应再通过L2正则方法交叉检验。

from sklearn.feature\_selection import SelectFromModel

#带L1和L2惩罚项的逻辑回归作为基模型的特征选择

#参数threshold为权值系数之差的阈值

SelectFromModel(LR(threshold=0.5, C=0.1)).fit\_transform(iris.data, iris.target)

###### 基于学习模型的特征排序

这种方法的思路是直接使用你要用的机器学习算法，针对每个单独的特征和响应变量建立预测模型。假如某个特征和响应变量之间的关系是非线性的，可以用基于树的方法（决策树、随机森林）、或者扩展的线性模型等。基于树的方法比较易于使用，因为他们对非线性关系的建模比较好，并且不需要太多的调试。但要注意过拟合问题，因此树的深度最好不要太大，再就是运用交叉验证。通过这种训练对特征进行打分获得相关性后再训练最终模型。

在波士顿房价数据集上使用sklearn的随机森林回归给出一个单变量选择的例子：

from sklearn.cross\_validation import cross\_val\_score, ShuffleSplit

from sklearn.datasets import load\_boston

from sklearn.ensemble import RandomForestRegressor

#加载波士顿房价作为数据集

boston = load\_boston()

X = boston["data"]

Y = boston["target"]

names = boston["feature\_names"]

#n\_estimators为森林中树木数量，max\_depth树的最大深度

rf = RandomForestRegressor(n\_estimators=20, max\_depth=4)

scores = []

for i in range(X.shape[1]):

#每次选择一个特征，进行交叉验证，训练集和测试集为7:3的比例进行分配，

#ShuffleSplit()函数用于随机抽样（数据集总数，迭代次数，test所占比例）

score = cross\_val\_score(rf, X[:, i:i+1], Y, scoring="r2",

cv=ShuffleSplit(len(X), 3, .3))

scores.append((round(np.mean(score), 3), names[i]))

#打印出各个特征所对应的得分

print(sorted(scores, reverse=True))

输出结果：

[(0.64300000000000002, 'LSTAT'), (0.625, 'RM'), (0.46200000000000002, 'NOX'), (0.373, 'INDUS'), (0.30299999999999999, 'TAX'), (0.29799999999999999, 'PTRATIO'), (0.20399999999999999, 'RAD'), (0.159, 'CRIM'), (0.14499999999999999, 'AGE'), (0.097000000000000003, 'B'), (0.079000000000000001, 'ZN'), (0.019, 'CHAS'), (0.017999999999999999, 'DIS')]

#### 参数调优

##### 网格搜索GridSearch

网格化寻优可以说是最基本的超参数优化方法。使用这种技术，我们只需为所有超参数的可能构建独立的模型，评估每个模型的性能，并选择产生最佳结果的模型和超参数。

##### 随机搜索RandomSearch

通常并不是所有的超参数都有同样的重要性，某些超参数可能作用更显著。 而随机寻优方法相对于网格化寻优方法能够更准确地确定某些重要的超参数的最佳值。

##### 贝叶斯调优HyperOpt/BayesOpt

前面两种方法能够针对单独超参数组合模型进行训练，并评估各自的性能。每个模型都是独立的，因此很易于进行并行计算。但是每个模型都是独立的，也导致模型之间不具有指导意义，前一模型的计算结果并不能影响后一模型的超参数选择。 而贝叶斯优化方法（顺序优化方法的一种，sequential model-besed optimization, SMBO）则可以借鉴已有的结果进而影响后续的模型超参数选择。

这也限制了模型训练评估的计算次数，因为只有有望提高模型性能的超参数组合才会被进行计算。

贝叶斯优化是通过构造一个函数的后验分布（高斯过程）来工作的，该后验分布最好地描述了要优化的函数。随着观测次数的增加，后验分布得到改善，算法更加确定参数空间中哪些区域值得探索，哪些区域不值得探索。

当反复迭代时，算法会在考虑到它对目标函数的了解的情况下，平衡它的探索和开发需求。在每个步骤中，高斯过程被拟合到已知的样本（先前探索的点），后验分布与探索策略（例如UCB（上置信限，upper confidence bound）或EI（预期改善, expected improvement））被用于确定下一个应该探索的点。

通过贝叶斯优化方法，可以更高效得探索超参数变量空间，降低优化时间。

# 竞赛实战

### 第三届阿里云安全算法挑战赛

[第三届阿里云安全算法挑战赛-天池大赛-阿里云天池 (aliyun.com)](https://tianchi.aliyun.com/competition/entrance/231668/information)

#### 问题背景

恶意软件是一种被设计用来对目标计算机造成破坏或者占用目标计算机资源的软件，传统的恶意软件包括蠕虫、木马等，这些恶意软件严重侵犯用户合法权益，甚至将为用户及他人带来巨大的经济或其他形式的利益损失。近年来随着虚拟货币进入大众视野，挖矿类的恶意程序也开始大量涌现，黑客通过入侵恶意挖矿程序获取巨额收益。当前恶意软件的检测技术主要有特征码检测、行为检测和启发式检测等，配合使用机器学习可以在一定程度上提高泛化能力，提升恶意样本的识别率。

本题目提供的数据来自文件（windows 可执行程序）经过沙箱程序模拟运行后的API指令序列，全为windows二进制可执行程序，经过脱敏处理。

本题目提供的样本数据均来自于从互联网。其中恶意文件的类型有感染型病毒、木马程序、挖矿程序、DDOS木马、勒索病毒等，数据总计6亿条。

#### 解决目标

数据说明

1. 训练数据（train.zip）：调用记录4亿次，文件11万个（以文件编号汇总），字段描述如下：



注1：一个文件调用的api数量有可能很多，对于一个tid中调用超过5000个api的文件，我们进行了截断，按照顺序保留了每个tid前5000个api的记录。

注2：不同线程tid之间没有顺序关系，同一个tid里的index由小到大代表调用的先后顺序关系。

注3：index是单个文件在沙箱执行时的全局顺序，由于沙箱执行时间有精度限制，所以会出现一个index上出现同线程或者不同线程都在执行多次api的情况，可以保证同tid内部的顺序，但不保证连续。

2）测试数据（test.zip）：调用记录近2亿次，文件5万多个。

说明：格式除了没有label字段，其他数据规格与训练数据一致。

#### 行动方案

##### 预处理

（1）去重：同一file\_id中，有大量重复数据记录

（2）清洗： return value的-1和4294967295其实是一样的取值，

还有些负值也有相应的正值出现，它们代表相同的意思

（3）优化：对api进行LabelEncoder、提取出label

##### 特征工程

###### 基础统计特征

(1) file\_id对每种api调用了多少次

(2) file\_id对每种返回值不为0的api调用了多少次

(3) file\_id调用了多少种api

(4) file\_id调用了多少个不同的tid及tid调用次数的max、min、mean

(5) file\_id调用后返回了多少种return\_value

(6) file\_id调用的最大index

(7) file\_id对调用api的最大index、最小index、平均index及调用阶段（index/length）

(8) 返回值为0的调用中,file\_id调用了多少个api

(9) 返回值为0的调用中,file\_id调用了多少个不同的tid

(10) 返回值为0的调用中,file\_id调用的最大index

(11)以上部分特征在细分tid粒度的基础上重新统计

###### 时序特征

1. file\_id对每种api的调用index间隔
2. file\_id对每个tid中的index调用间隔

###### N-gram特征

1. 先数一遍每个api的return value有多少种，小于等于20的保留作为类别，大于二十的就以大于零和小于等于零分成两类。 然后把api和return value合成一列，做2-gram的count，一共6万多的特征，然后跑一个lgb按importance筛掉，剩1700+特征。
2. 1-gram至5-gram的api，同时将连续调用同样的api去重后再进行计算。

###### label特征

用模型进行二分类预测得到的概率，做法是将异常文件（label!=0）作为正样本进行二分类预测，训练集划为五折，进行交叉预测二分类概率，测试集取五折预测的平均值，该特征线上可提升五个千分点

###### 性能优化

1. 将api进行LabelEncoder 编码 ，减少内存占用，加快提特征的速度。
2. 训练集分成三份，按行读取，多线程提取特征，分块存储。

##### 模型选择LightGBM

###### 策略选择

1. GOSS（从减少样本角度）：排除大部分小梯度的样本，仅用剩下的样本计算信息增益。
2. EFB（从减少特征角度）：捆绑互斥特征，也就是他们很少同时取非零值（也就是用一个合成特征代替）。
3. gbdt是基于决策树的集成算法，采用前向分布算法，在每次迭代中，都是通过负梯度拟合残差，从而学习一颗决策树，最耗时的步骤就是找最优划分点。Lightgbm采用直方图算法，他把连续特征值划分到k个桶中取，划分点则在这k个点中选取。

###### 模型特点

1. 基于Histogram的决策树算法
2. 带深度限制的Leaf-wise的叶子生长策略
3. 直方图做差加速
4. 直接支持类别特征
5. Cache命中率优化
6. 基于直方图的稀疏特征优化
7. 多线程优化

###### 模型优势

1. 更快的训练效率
2. 低内存使用
3. 更高的准确率
4. 支持并行化学习
5. 可处理大规模数据

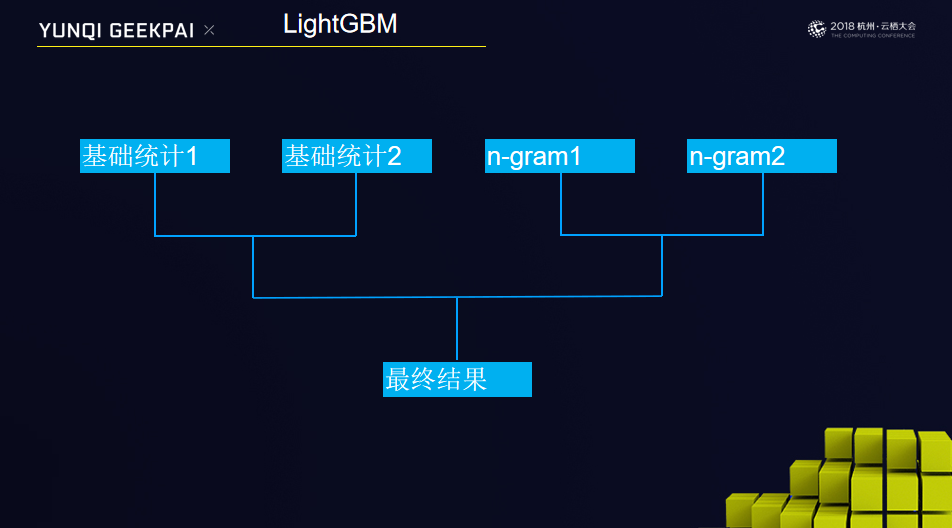
##### 模型调优HyperOpt

局部参数调优：对模型效果影响较大的有叶子数、列采样比例、L2正则

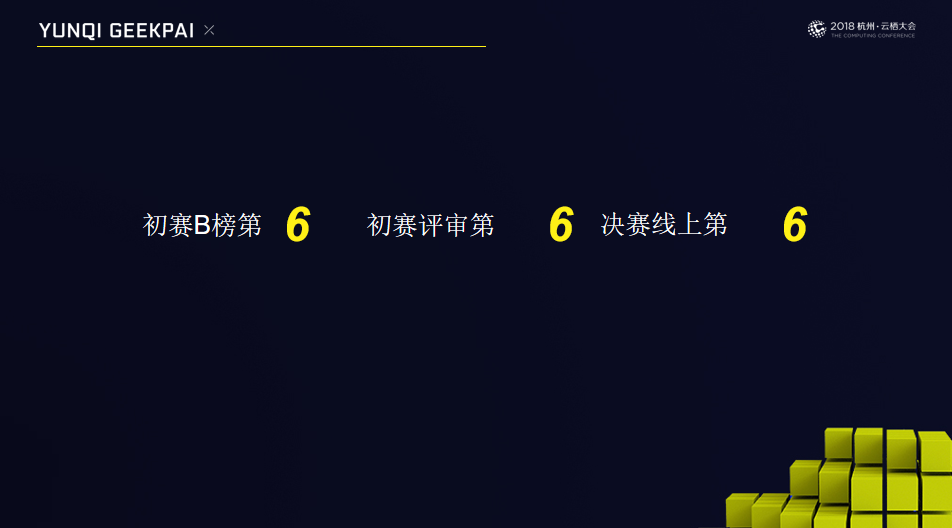
##### 模型融合根据相关系数进行加权平均







#### 达成结果



### **OPPO搜索排序OGeek算法挑战赛**

[OGeek算法挑战赛赛题与数据-天池大赛-阿里云天池 (aliyun.com)](https://tianchi.aliyun.com/competition/entrance/231688/introduction)

[TIANCHI天池-OGeek算法挑战赛分享及完整代码（亚军） - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/51422621)

#### 问题背景

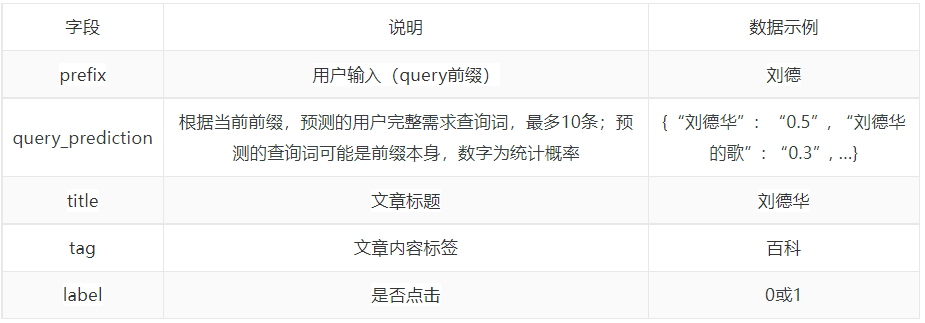
在搜索业务下有一个场景叫实时搜索（Instance Search）,就是在用户不断输入过程中，实时返回查询结果。  
  
此次赛题来自OPPO手机搜索排序优化的一个子场景，并做了相应的简化，意在解决query-title语义匹配的问题。简化后，本次题目内容主要为一个实时搜索场景下query-title的ctr预估问题。本次赛题为开放型算法挑战赛，优秀的解决方案会对我们解决这个场景以及其它场景下的问题带来极大的启发。我们期待优秀的你和你的团队能够投入进来！  
  
OPPO场景：  


#### 解决目标

**题目内容**

基于百万最新真实用户搜索数据的实时搜索场景下搜索结果ctr预估。  
给定用户输入prefix（用户输入，查询词前缀）以及文章标题、文章类型等数据，预测用户是否点击。文章资源类别非全网资源，属部分垂直领域内容。初赛后期开放B榜开放时间3小时，请在三小时内提交结果；初赛结束时需要提交完整代码，最终晋级复赛前100名；复赛全程采用线上赛形式。

**数据描述**此次初赛数据约235万 训练集200万，验证集5万，A榜测试集5万，B榜测试集25万，数据全部来源于OPPO日常搜索真实用户点击数据，数据集内可能存在重复、矛盾、同一搜索词对应多个点击等都是真实存在的。  
  
数据格式： 数据分4列，\t分隔



**初赛数据量：**

训练数据：200万

验证数据：5万

测试数据1：5万

测试数据2：25万

**样例数据：**

挂号 {"挂号信是什么": "0.023", "挂号网上预约": "0.029", "挂号网官网": "0.015", "挂号信": "0.082", "挂号": "0.066", "挂号信单号查询": "0.075", "挂号平台": "0.025", "挂号网": "0.225", "挂号信查询": "0.201", "挂号信查询中国邮政": "0.020", "挂号预约": "0.021"} 预约挂号网 应用 1

挂号 {"挂号信是什么": "0.023", "挂号网上预约": "0.029", "挂号网官网": "0.015", "挂号信": "0.082", "挂号": "0.066", "挂号信单号查询": "0.075", "挂号平台": "0.025", "挂号网": "0.225", "挂号信查询": "0.201", "挂号信查询中国邮政": "0.020", "挂号预约": "0.021"} 挂号网 网站 0

文本编码格式：UTF-8

**复赛说明**

复赛数据包括1个数据包和4个词向量。oppo\_data\_round2\_20181107包含此次复赛所需的训练集，验证集和测试集。四个文件（merge\_sgns\_bigram\_char300.tar.gz、sgns.merge.char.tar.gz、sgns.merge.word.tar.gz、sgns.merge.bigram.tar.gz）为词向量，这四个词向量均为同一份综合语料通过不同方法训练得到的四种预训练词向量，参见https://github.com/Embedding/Chinese-Word-Vectors。

复赛平台使用和要求。（链接）

**复赛要求：**

1）除停用词等小文件外不得上传第三方数据。

2）模型数量不多于2个（说明链接）；

3）最终代码在平台上时间限定为CPU方案8小时，GPU方案16小时内完成训练和预测。两者都采用的方案总时间限制为16小时。

4） 禁止人工标注/修改评测结果数据；

5）复赛平台不允许上传及运行与比赛无关的数据和代码，一经发现收回账号权限。

**结果提交说明**

本次竞赛要求选手按照测试集顺序生成 csv 结果文件，压缩成 zip 文件提交，文件名不限。每行是一个预测结果的标签值，不需要表头，示例如下：

0

1

1

**评分标准**

本次竞赛的评价标准采用F1 score 指标，正样本为1。计算方法参考https://en.wikipedia.org/wiki/F1\_score

**注意事项**

为了更加贴近真实情况，测试集中完全相同的词条有很多，证明该词条在真实环境中的搜索比例比较大，在评测中相应权重应该较大，体现在出现频率较高。

此赛题需要参赛者注意以下行为：

1） 禁止人工标注/修改评测结果数据；

2）模型数量限制不多于2个；

3）除训练词向量和预训练的词向量外禁止使用第三方数据，预训练词向量需注明出处和链接，训练词向量需提交完整脚本；

4）禁止多账号刷分等。

关于模型数量限制的补充说明：

比赛中，使用多个模型的结果进行简单融合，被认为是多模型策略。我们不鼓励过度堆砌模型和硬件来刷高比赛得分的行为。为倡导比赛算法的创新性和实用性，我们在大赛过程中对“多模型策略”进行限制。分2个阶段进行说明：

a)预测阶段

大赛要求预测阶段仅能使用不超过2个的模型。

注意：同一网络结构但参数不同也被认为是不同模型。

b)预处理阶段

我们认定预处理阶段的数据处理中使用到word2vec等词向量模型不计入模型数量，其他有不确定的情况可以到技术圈提问。

#### 行动方案

##### 赛题回顾

意义：此次赛题来自OPPO手机搜索排序优化的一个子场景，并做了相应的简化，解决query-title语义匹配的问题

问题转化：实时搜索场景下query-title的ctr预估问题

赛题特征：prefix,query\_prediction,title,tag

正负样本比例：6:10

数据规模：初赛：训练集200w 验证集5w 测试集一5w 测试集二25w

复赛：

评估指标：F1 score 指标

##### 数据初探

从value\_counts,点击率,分布情况进行了解

##### 数据预处理

query\_prediction移除标点符号并构造query\_prediction key

##### 特征工程

基础特征、文本处理特征、一阶统计特征、二阶交叉特征、CountVectorizer特征、TF-IDF特征

###

对query\_prediction列表,prefix,title字符串求长度

prefix是否存在title中

title与prefix长度差,比

prefix在query\_prediction key中出现次数

title在query\_prediction key中出现次数

获取query\_prediction中有title出现的最大统计概率

获取prefix,title结巴分词后的长度

获取query\_prediction 中统计概率小于0.08的个数

title是否在query\_prediction中,在其中的位置,统计概率

### 构造SequenceMatcher相似度特征

prefix,title分别构造与query\_prediction中每个key的相似度,并统计max,std,mean相似度,

统计top3,top5相似度的max,std,mean。

对query\_prediction中的统计概率也做以上统计

prefix和title的共现词

计算查询词prefix出现在title中的那个位置，前、后、中、没出现

### 常规统计特征

Ctr : 一阶二阶,做平滑处理

Ratio : 一阶二阶

Count : 一阶二阶

Unique : 一阶二阶

### 词频统计CountVectorizer

对prefix,title,query\_prediction 进行jieba分词

构造CountVectorizer特征

提取prefix对应的所有tag,然后对其构造CountVectorizer特征

##### 算法模型

使用lightgbm模型,GBDT模型记忆性更强，记忆特征和标签相关特征组合能力强，同时更具

可解释性，稳定。

##### 模型融合

两组具有差异性特征训练出来的Lightgbm模型，最终根据线上得分进行加权融合

##### 总结

模型简单稳定,训练时间短,三个小时即可训练完。

深入挖掘特征,从传统统计特征到语义特征,刻画不同关键词的重要性。

#### 达成结果

1.解题思路

模型数量:2个lightgbm模型的简单加权融合, 最终结果由两套代码生成结果加权融合所得, 两个结果均有LightGBM训练所得, 不同的是特征的差异性.

参数的差异性。

主要特征:

a.基础特征:

1).主要包括prefix/title/max\_query\_prediction\_keys相似度的系列特征, 对于一系列的统计list的数学统计特征, 分别统计list中的最大/最小

均值/标准差/众数等等;

2).转换率特征, 关于prefix/title/tag的相关转换率以及组合的一些转换率, 统计的时候用了一些简单的sample trick, 防止过拟合;

3).点击特征, 统计关于prefix/title/tag的单个点击以及组合的点击特征;

4).统计nunique特征, 例如prefix下有多少个title等等类似的特征;

b.countvector特征:对于prefix/title/query\_prediction用cv变换成词的稀疏矩阵输入进模型一起训练, 另外还有prefix下包括哪些tag字符串的countvector;

2.哪部分是开源的

在utility文件中, 有一些功能函数, 例如计算两个字符串间的各种相似度, 以及转换率平滑函数HyperParam, 都是开源的

### **2018科大讯飞人工智能营销算法大赛**

[DC竞赛-大数据竞赛平台 (datacastle.cn)](https://challenge.datacastle.cn/v3/cmptDetail.html?id=245)

[2018科大讯飞AI营销算法大赛总结及完整代码（冠军） - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/47807544)

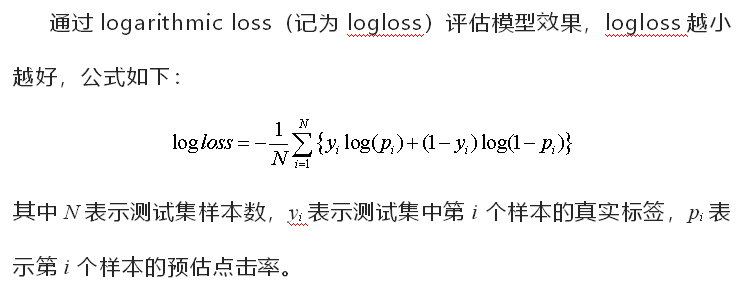
#### 问题背景

讯飞AI营销云在高速发展的同时，积累了海量的广告数据和用户数据，如何有效利用这些数据去预测用户的广告点击概率，是大数据应用在精准营销中的关键问题，也是所有智能营销平台必须具备的核心技术。

本次大赛提供了讯飞AI营销云的海量广告投放数据，参赛选手通过人工智能技术构建预测模型预估用户的广告点击概率，即给定广告点击相关的广告、媒体、用户、上下文内容等信息的条件下预测广告点击概率。希望通过本次大赛挖掘AI营销算法领域的顶尖人才，共同推动AI营销的技术革新。

#### 解决目标

**讯飞AI营销广告点击率预估，预测广告被点击的概率。**



数据说明

提供下载的数据集包括两个部分：

初赛：

1. round1\_iflyad\_train.txt 训练集，每一行数据为一个样本，可分为5类数据，包含基础广告投放数据、广告素材信息、媒体信息、用户信息和上下文信息，共1001650条数据。其中‘click’字段为要预测的标签，其它34个字段为特征字段。 2. round1\_iflyad\_test\_feature.txt 测试集，共40024条数据，与训练集文件相比，测试集文件无‘click’字段，其它字段同训练集。

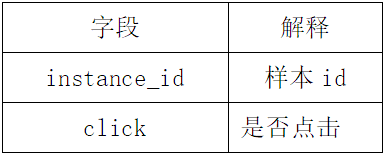
复赛：

1. round2\_iflyad\_train.txt 训练集，每一行数据为一个样本，可分为5类数据，包含基础广告投放数据、广告素材信息、媒体信息、用户信息和上下文信息，共1998350条数据。其中‘click’字段为要预测的标签，其它34个字段为特征字段。 2.round2\_iflyad\_test\_feature.txt 测试集，共80276条数据，与训练集文件相比，测试集文件无‘click’字段，其它字段同训练集。

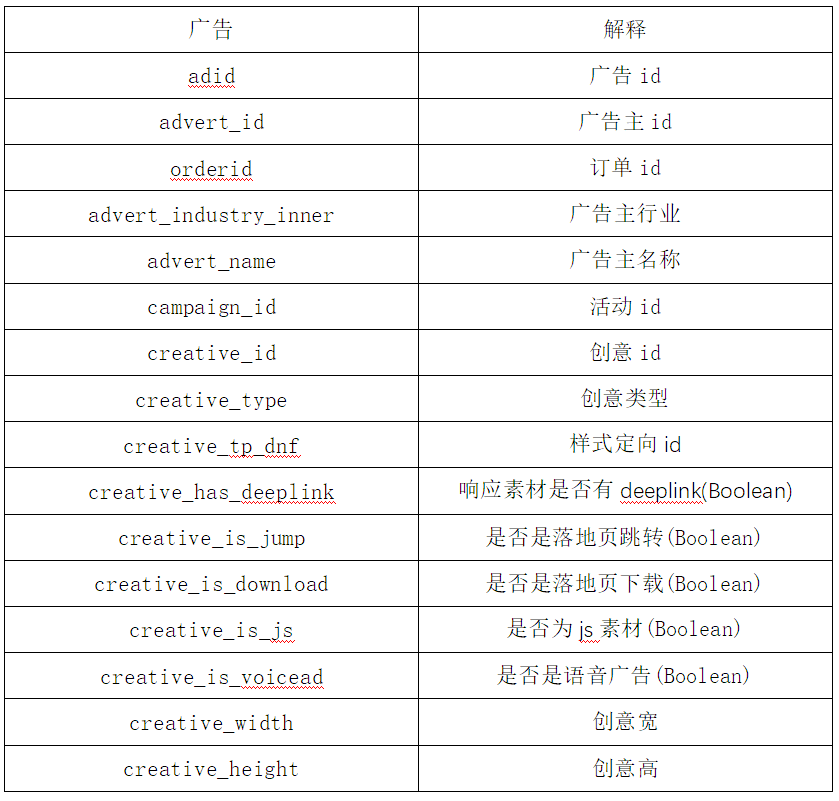
出于数据安全保证的考虑，所有数据均为脱敏处理后的数据。数据集提供了若干天的样本，最后一天数据构成了测试集，其余日期的数据作为训练数据。

注意：此次比赛分为初赛和复赛两个阶段，两个阶段的区别是所提供样本的量级有所不同，其他的设置均相同。 5类数据的描述如下：

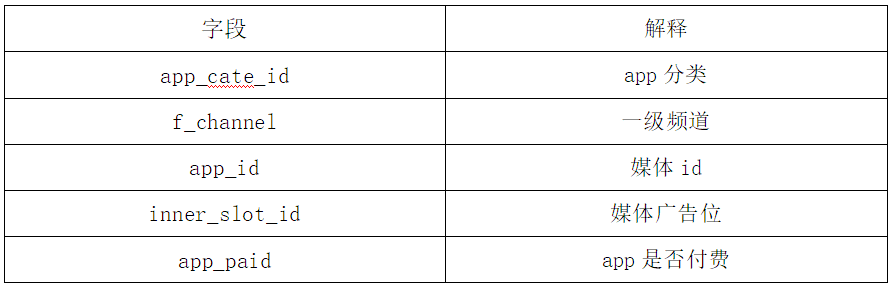
基本数据:



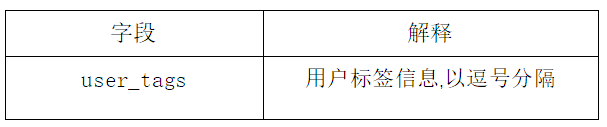
广告信息:



媒体信息：



用户信息：



上下文信息：



补充说明： 1.advert\_industry\_inner字段数据样例为102400\_102401，“102400(前者)”表示广告主一级行业标签id，“102401(后者)”表示广告主二级行业id，如“教育\_培训” 2.time字段脱敏后为有序排列，且时间间隔和与真实时间对应。

#### 行动方案

[[金山文档] 酒水鱼1.0.0.pptx](https://kdocs.cn/l/cbT2jhPARg2x)

算法说明：

##### liupeng\_demo.py

预处理（数据清洗）>--特征工程(统计组合)>--Stacking减少总特征维度（LightGBM）>--训练预测

1、预处理

（1）初复赛训练数据合并后去重

（2）提取广告投放信息，日期、小时以及早中晚时间段

（3）细分广告主行业与媒体广告位，去除只有一个取值的字段（三个Boolean字段）

（4）清洗手机品牌和机型字段，对同类进行合并

（5）对操作系统及其版本、名称进行更细粒度的刻画

（6）构造虚拟用户组别，对具有相同标签的用户进行统一编码

（7）对除用户标签以外的类别特征进行独热编码

2、特征工程

（1）用户特征提取：

对用户标签进行CountVectorizer之后利用LightGBM特征重要性提取前100维的标签

（2）主要特征组别：

一维+二维count计数特征（如广告主id共计投放次数）

类别偏好的ratio比例特征(如广告主id的某个广告id投放比例)

类别变量的nunique特征（如广告主id有多少个不同的广告id）

用户标签与其他字段的组合mean特征（如广告id对用户性别的投放比例）

（3）特征处理方式：

由上可知组合特征维度非常大，本次竞赛采用Stacking的办法将组合特征的信息进行降维压缩，最后总计使用特征

只有一百五十多维

##### wanghe\_demo.py

预处理（数据清洗）>--特征工程(统计组合)>--训练预测

1、预处理

（1）初复赛训练数据合并后去重

（2）提取广告投放信息，日期、小时

（3）细分广告主行业与媒体广告位，去除只有一个取值的字段（三个Boolean字段）

（4）清洗手机品牌和机型字段，对同类进行合并

（5）对操作系统及其版本、名称进行更细粒度的刻画

（6）构造虚拟用户组别，对具有相同标签的用户进行统一编码

（7）构造类别间交叉特征，获取更细粒度的信息

（8）对除用户标签以外的类别特征进行Labelencoder

2、特征工程

（1）用户特征提取：

对用户标签进行CountVectorizer之后利用卡方检验进行特征选择

（2）主要特征组别：

对交叉特征进行计数统计构造count特征（如广告主id共计投放次数）

类别偏好的ratio比例特征（如广告主id的某个广告id投放比例）

类别变量的nunique特征（如广告主id有多少个不同的广告id）

类别变量的点击率特征（考虑到信息泄露带来的过拟合问题，构造了历史转化率特征）

（3）特征处理方式：

ratio比例特征的经过lightgbm返回的特征重要性,获取重要性大于10的ratio特征

##### wengyp\_demo.py

预处理 >--特征工程>-->训练预测

1、预处理

（1）初复赛训练数据合并后去重

(2）提取广告投放信息，日期、小时

(3) 当water\_demo.py中 PROCESSED\_FILE 设置为False时，对user\_tags进行处理，保留在训练集中出现次数排在较前的tags（目前选取前399个）

处理后的新训练集和测试文件保存在 data/processed\_train\_fusai2.csv 和 ./processed\_test\_fusai2.csv

如果进行过预处理并保存了文件，则PROCESSED\_FILE设置为True,可以大大提高速度，预处理操作只需要进行一次,后续可直接使用处理过的文件。

（3）细分广告主行业与媒体广告位

（4）清洗手机品牌和机型字段，对同类进行合并，对操作系统及其版本、名称进行更细粒度的刻画(参考队友刘鹏的操作）

（5）对原始测试集中出现的类别特征，且在原始测试集中的类别数目小于50的特征进行独热编码（参考比赛群内公开的代码）

2、特征工程

（1）用户特征提取：

采用将预处理后的用户特征编码为向量格式，每个用户保留399维的用户向量特征，其中第i维代表第i个tag出现的个数

（2）主要特征组别：

计数特征：对部分类别特征计算出现在测试集和训练集中的总数，对媒体特征，部分手机属性和上下文属性进行拼接，进行计数。

统计了部分类别特征在对应曝光记录的训练集中前一天中出现的次数。

组合特征： 对媒体特征，部分手机属性和上下文属性进行拼接,组合作为字符串进行labelEncoder操作加入特征

分段Ratio特征：

计算部分类别型特征在不同时间段的点击率，包括该曝光前所有数据关于该特征的聚合点击率，以及该曝光发生前一天针对该特征的

聚合点击率。

3、模型：

最后采用单个lightGBM模型，将训练集分为五折进行训练和验证，将各次对测试集的预测结果进行平均即可。

##### 总结

liupeng\_demo.py:

(1) 建模过程中充分考虑了用户标签与其他信息的交互作用（mean特征），并采用Stacking抽取特征信息的

(2) 方式减少维度与内存的使用，对广告与用户交互信息的充分挖掘，也使得模型在AB榜测试相对稳定

wengyp\_demo.py:

(1) 对特征处理时考虑了不同时间尺度的历史点击率信息，既利用了历史曝光信息和点击率信息，又能减少过拟合的风险。

(2) 对不同类型特征的组合可以增强表达能力，提高预测的效果。

(3) 对用户标签的筛选可以减少特征的维度并减少噪声。

wanghe\_demo.py:

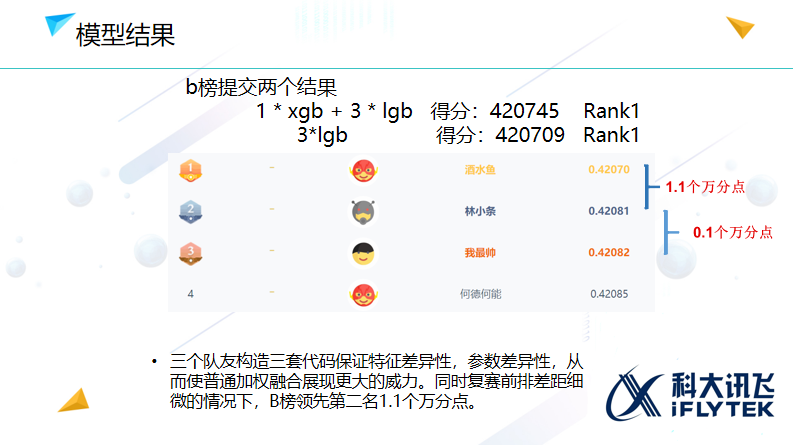
(1) 树模型Memorazation更强，记忆特征和标签特征相关特征组合能力强，可以轻松驾驭小数据集比赛，并且有很好的效果。

(2) 获取更细粒度的表现信息进行交叉特征组合，在推荐类别的场景达到精准推荐的作用。

(3) 因为user\_tags多值特征包含用户的属性信息，所以完美的表达user\_tags至关重要，构造特征中不仅考虑了对其进行展开，而且对其中重要属性特征进行扩展，接近完美的表达user\_tags。

(4) user\_tags,f\_channl缺失值虽然多，但通过数据分析其转化率，可以清楚的看出其区分度。

#### 达成结果



### **2018腾讯广告算法大赛**

[2021腾讯广告算法大赛 (qq.com)](https://algo.qq.com/archive.html?)

[2018腾讯广告算法大赛总结/0.772229/Rank11 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/38034501)

[[金山文档] 第十五章 Tencent实战案例\_终稿.docx](https://kdocs.cn/l/cqBl5XvfEoRk)

### 第三届融360天机智能金融算法挑战赛

[第三届融360天机-智能金融算法挑战赛多场景金融赛题--TOP1方案 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/51927257)

#### 问题背景

#### 解决目标

#### 行动方案

#### 达成结果

### Kaggle竞赛平台

#### 问题背景

#### 解决目标

#### 行动方案

#### 达成结果

### 机器学习算法竞赛实战

[[金山文档] 第一章 竞赛介绍\_终稿.docx](https://kdocs.cn/l/csA7nExCu1GB)

[[金山文档] 第七章 用户画像\_终稿.docx](https://kdocs.cn/l/ca9Y36q96FDd)

[[金山文档] 第八章 Elo实战案例\_终稿.docx](https://kdocs.cn/l/cfnJdf6tmMR7)

[[金山文档] 第十五章 Tencent实战案例\_终稿.docx](https://kdocs.cn/l/csAoPYmZZFnZ)

[[金山文档] 第十三章 计算广告\_终稿.docx](https://kdocs.cn/l/cjci9GkP7hcs)

#### 内容介绍

对于本书的章节架构，我们除了进行仔细的讨论外，还采纳了国内多名顶尖竞赛选手的建议。

算法竞赛本身涵盖的范围是很大的，我们的理念是剖析其最本质的内容，然后结合多个领域模

块进行实战讲解，这也是本书的一大特色。本书分为以下五个部分。

第一部分——磨刀事半，砍柴功倍。这部分以算法竞赛的通用化流程为主，介绍竞赛中各

个部分的核心内容和具体工作，且每章都配有具体的实战部分，以便加深理解。

第二部分——物以类聚，人以群分。这部分主要介绍用户画像相关的问题，构建完善的标

签体系是用户画像的核心，也是解决用户画像类赛题的关键，比如个性化推荐和金融风控等问

题都需要以用户画像作为支撑。为了帮助读者加快对此类竞赛问题的学习、理解，会讲解具体

的竞赛案例，即 Kaggle 平台的 Elo Merchant Category Recommendation。

第三部分——以史为鉴，未来可期。这部分以时间序列预测问题为主，先讲述这类问题的

常见解题思路和技巧，然后分析两个具体的实战案例，分别是天池平台的全球城市计算 AI 挑战

赛和 Kaggle 平台的 Corporación Favorita Grocery Sales Forecasting。

第四部分——精准投放，优化体验。计算广告相关的业务大都是很好的竞赛题目，这

部分主要介绍了计算广告的核心技术和业务，包括广告召回、广告排序和广告竞价。实战案

例部分则包括两道赛题，分别是 2018 腾讯广告算法大赛——相似人群拓展和 Kaggle 平台的

TalkingData AdTracking Fraud Detection Challenge。

第五部分——听你所说，懂你所写。这部分基于自然语言处理相关的内容进行讲解，包括

常见任务和常见技术，实战案例部分是 Kaggle 平台上的经典竞赛 Quora Question Pairs。

本书是算法竞赛领域第一本系统性介绍竞赛的书籍，不仅包含竞赛的基本理论知识，还结

合多个方向和案例详细阐述了竞赛中的上分思路和技巧。

# 工作经历

## 中国电信

**北京研究院 网络研发与运营支持部 机器学习实习生**

* 分析4Gqos提速对普通用户的感知影响，并建立模型预测其是否劣化，最终精度提高了45%。
* 分析4Gqos迅游对VIP用户的提速效果，挖掘用户历史行为与基站特征并建立模型预测提速是否有效。

### 数据分析

jupyter notebook

### 数据挖掘

### 数据建模

## 华为技术有限公司

**运营商BG 全球技术服务部（GTS）服务与研发管理部 算法与技术开发部**

### TeleAutoML

#### 问题背景

在电信领域，运营商手里存在大量的用户、小区与基站相关的数据与应用场景，很多都有可能使用到可解释性较强的机器学习进行建模。华为在给国内外各大运营商客户提供硬件设施的同时，也提供相应的网络运维与网路故障诊断服务，此外，运营商有一些涉及到AI的业务需求也会考虑寻求这方面的支持，因此针对客户的普遍需求，与场景数据的特点，自研可以快速准确响应客户需求的电信领域自动机器学习TeleAutoML组件。

#### 解决目标

自研开发适配电信智能运维领域多种应用场景的TeleAutoML，如流量预测、拥塞预测、用户留存、离网预测、用户体验、用户提速、套餐推荐、用户标注以及其他运营商客户有需要的业务场景。

#### 行动方案

建立端到端的电信领域自动化机器学习建模方案，包含以下功能：

* 支持傻瓜式训练与预测
* 支持分类与回归两种有监督学习方法
* 支持LR/RF/GDBT/SVM/LGB/XGB
* 支持时间序列建模
* 支持离散类别特征
* 支持文本类特征
* 支持连续特征
* 支持特征组合
* 支持特征选择
* 支持并行加速
* 支持模型选择
* 支持参数调优
* 支持交叉验证
* 支持多种模型评价
* 支持模型融合
* 支持模型集成
* 支持Python/Java/Scala接口调用
* 支持命令行调用

#### 达成结果

形成内部开源包，与相应的指导安装与操作手册，供一线人员快速响应客户建模需求与场景诉求。

### 用户画像eUEBA

#### 问题背景

在拥有TeleAutoML的能力基础上，鉴于运营商客户有很多应用场景都是面向其移动端用户的，因此对于用户的行为刻画对于各类场景的建模效果尤为重要，因此需要分析采集数据源的数据类型搭建相应的用户画像特征工程。

#### 解决目标

尽可能在刻画用户群体行为的同时最大程度区别出用户个体的差异性，从而接近真实的用户行为逻辑与所处情形，判断其使用痛点、预测其未来行为、提供更好体验的同时，为运营商客户带来相应的商业利益。

#### 行动方案

根据用户的网络使用记录，进行相应的用户画像搭建，数据源主要有记录了用户详细移动流量使用情况的XDR与SDR、用户缴费与套餐余额信息、用户通话与短信信息，可以得出用户的以下行为画像：

* 用户的时空轨迹
* 用户的套餐资费、超套使用
* 用户的社交联络信息
* 用户的上网习惯、时间、地点、时长、内容、偏好

#### 达成结果

在东南亚泰国、缅甸局点的APP推广、用户离网预测、年龄段识别、套餐推荐、预付费充值等项目建模效果获得客户认可。

### 智能运维AIOps

#### 问题背景

4G/5G网络由基站、小区、用户之间的交互传递信息，在资源有限、硬件设备物理受限、软件调优具有不确定性的情况之下，时常会发生网络故障告警异常，给用户的上网与通信体验造成极坏的影响，同时给运营商客户带来口碑与流量的损害。

#### 解决目标

根据告警或异常找出相关联的小区，排查故障的同时，评估其造成的全网流量损失，识别相关指标的互相影响关系，在未来呈现类似趋势时提供相应的预警。

#### 行动方案

时间序列建模、相关性分析、异常检测。

#### 达成结果

评估告警的影响范围，与流量损失，在下次遇到类似事件时进行提前预警。

### 因果推理Caufer

#### 问题背景

在通信4G/5G领域，存在的很多故障场景，需要对上报期间的日志进行定位分析，识别与现象相对应的根因。

#### 解决目标

从原始日志解析数据进行节点表征学习，然后使用信念传播和贝叶斯网络进行推理，给出根因叶子节点的概率排序以及对应的故障传播因果链。

#### 行动方案

1. 知识图谱与业务专家构图
2. 原始日志解析数据预处理
3. 参数判断与节点表征学习
4. 因果证据推理叶子节点
5. 根因与传播链分析

#### 达成结果

在多个场景下，TOP3推理准确率达85%以上。

# 技能证书

## GitHub

https://github.com/liupengsay

## LeetCode

[力扣 (leetcode-cn.com)](https://leetcode-cn.com/profile/)

## 专业能力

* 大学英语六级。
* 熟练使用**Python**语言进行数据分析、特征提取以及机器学习建模。
* 了解SQL、Scala语言，熟练使用**Git、Pycharm**进行工程化开发测试。
* 掌握scikit-learn、LightGBM、XGBoost、CatBoost等算法包。
* 华为可信认证Python专业级（科目一可信上机编程、科目二可信编程知识与应用、科目三可信软件工程与技术、科目四可信设计与重构）

# 自我评价

* 为人踏实沉稳，友好善良，真诚谦虚，勤奋刻苦。学习成绩优秀，本科期间3年累计获奖学金2万，有良好的数理基础以及逻辑分析能力。工作积极主动，沟通能力强，有较强的团队精神和强烈的责任心，对互联网和机器学习感兴趣，在机器学习方面有较为丰富的比赛经历与实践经验。热爱生活、喜欢摄影和健身，具有自律精神和自我驱动。