可能是全网写特征工程最通透的...

原创 包包闭关修炼 包包算法笔记 2021-12-28 08:41

收录于话题
#深度学习 11 #机器学习 11 #特征工程 2

特征工程到底是什么？今天是包大人出租车原创日更，因为正在出差去机场的路上...

前言

现在这个知乎上的回答，包大人那个还是处于排序第一的位置，在知乎上有**158万** 浏览， **2K+赞**， **6K+收藏**，**2专业认可**。

这在一个专业的问题上，这个数据其实还是不容易的。类比知乎上的一个问题，是明白了什么让你编程水平突飞猛进的。从我个人来讲的话，理解了特征工程的精髓，让我机器学习水平突飞猛进。今天旧瓶装新酒，再重答一下，用最白话的讲最深刻的道理。

从一道题讲起

当时很多回答的毛病在于侧重于**"术"**，而忽略了**"道"**。讲了很多非常细节的操作方法，甚至把特征工程狭义理解成了数据的变换方法。

这里毛病很大的。

上来先出一道题目。

题目：请使用一个逻辑回归的模型，建模一个身材分类器，身材分偏胖和偏瘦两种，输入的特征有身高和体重。

这时候你发现，这个问题不是那么好**“线性”**解决的，线性解决的意思就是我拍两个系数加权，使用 sigmoid(ax+by+c)就搞定了。

事实上，我们很难单纯地从身高和体重决策出一个人的身材，你说姚明体重280斤，他真的一定就胖吗？？别忘了他身高有226公分的。这组数据可能超出了你的认知，只看数据不看照片，一下子不好说他是胖还是瘦。（其实挺胖的哈哈）

嗯，这个你看到那组数据，不好一下子说出来的感觉，**就是机器学习里面非常关键的概念，“非线性”。**

那么我们怎么解答这个问题呢？

方法有两个：
1.升级模型，把线性的逻辑回归加上kernel来增加非线性的能力。我们使用这个模型 sigmoid(ax+by+kx*y^(-2)+c)，这个模型通过多项式核方法的升级，解决了低维空间线性模型不太好解决的问题。
2.特征工程，掏出体检报告上的**BMI指数**，**BMI=体重/(身高^2)**。这样，通过BMI指数，就能非常显然地帮助我们，刻画一个人身材如何。甚至，你可以抛弃原始的体重和身高数据。

好了，现在你大概对特征工程有点眉目了。

方式一在理论上对应的东西就是提升VC维，方式二就是让你dirty hand的特征工程。

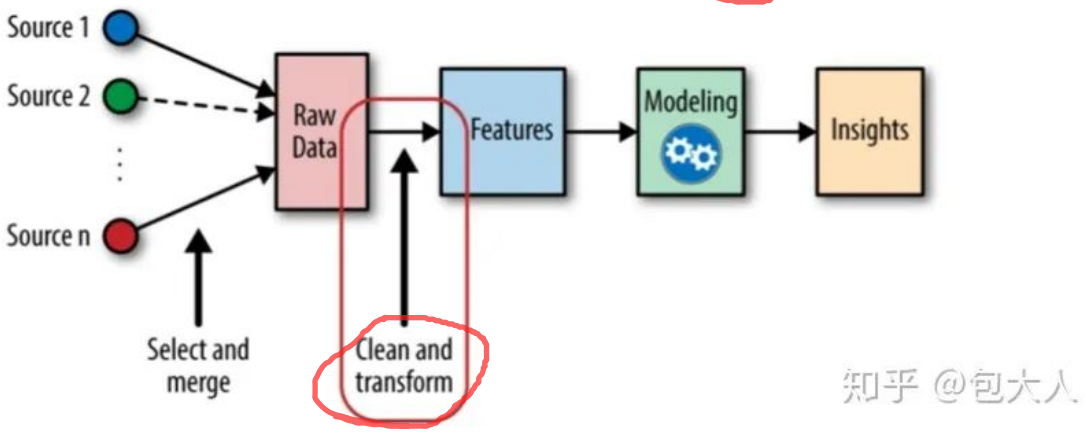
在机器学习流程中的视角

我们再回头讲一下这个东西的重要性，特征工程是机器学习，甚至是深度学习中最为重要的一部分，也是课本上最不愿意讲的一部分，特征工程往往是打开数据密码的钥匙，是数据科学中最有创造力的一部分。因为往往和具体的数据相结合，很难优雅地系统地讲好。所以课本上会讲一下理论知识比较扎实的归一化，降维等部分，而忽略一些很dirty hand的特征工程技巧，和case by case的数据理解。

不妨我们再讲透一点，下面是一个经典的特征机器学习流程，输入的原始数据经过一步clean and transformer转化为建模的特征输入，最终得到模型。

建模就是从数据中学习到insights（洞见）过程，这个过程其实是很曲折的，他要经过数据的表达，模型的学习两步。

数据的表达就是原始数据经过clean and transformer得到features的过程，即为特征工程。



还是以具体的例子来讲。

我们回到刚才的身材分类器的例子上，在方式一我们使用了核方法给逻辑回归升维，方式二使用了特征方法。

要知道天下没有免费的午餐，在你使用核方法升维的时候，实际很难精炼出恰好是x*y^(-2)这样的多项式表达，你肯定是一股脑地把x*y，x^2*y, x*y^2 这些项都扔进去了。

这么暴力的操作，有两个问题，一是共线性，二是噪声。

第一、共线性的意思是几个项表达的含义是趋同的，保持了很强的线性关系，对于逻辑回归是致命的问题，因为他带来了权重的不稳定，要知道逻辑回归权重可是暗示了特征重要性的。
（要是你对这段话，不好理解的话，仔细学习下逻辑回归模型和共线性的理论，此处不单独展开）

第二、噪声让你的分类器学习到了一些不好的东西，对你的决策没有产生泛化的贡献，反而带跑偏你的模型，学习到了一些不是知识的边边角角。

特征工程类似于炼丹术士的精炼过程。

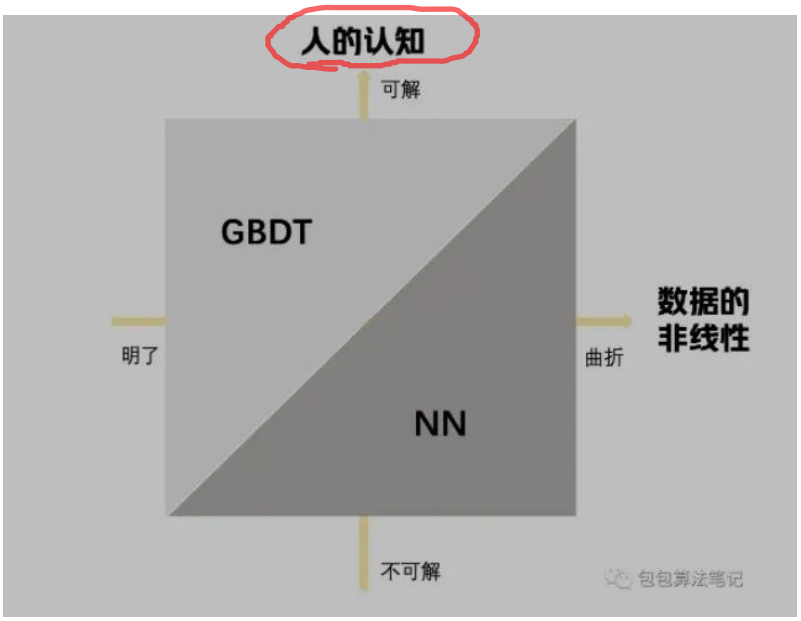
他的作用就是把人的知识融入到数据表达中，减轻模型的负担，让模型更容易学习到本质的知识。

从NN与GBDT讨论的视角

之间写过一篇文章，为什么GBDT可以超越深度学习，有读者说我是标题党，我结合这篇推文，继续把他翻出来。
那篇文档里有个很重要的观点是这张图。

这个图表达意思是，在人的认知可解的时候，乃至超越数据的非线性，就是y>x那根斜对角线的上方，是GBDT可以超越神经网络的时候。

人的认知可解，对应着就是特征工程的难度和可行性。



还是不好理解，看下面这样一个具体的例子。

如果你对speech稍有了解，或者做过说话人验证/声纹识别（SVR）任务，你会知道，有一种特征工程叫做MFCC 特征，现在解决说话人本身特性的问题，前端还是无法离开MFCC，而我认为MFCC是一种非常有代表性的饱含了专家知识的特征工程，感兴趣的同学可以了解一下相关的知识。
如果你有了MFCC这样的工具，你用不太复杂的浅层NN（TDNN），都能取得超越输入原始语音采样序列，放到复杂transformer中的效果。

另外，在某些类型的数据上，特种工程很难施展拳脚，类似于id序列，典型的任务如NLP，CTR等。

NLP是人类知识凝练的字节单元，CTR是大规模的稀疏id序列，这两种数据上，虽然人的认知是到位了，但是不可解。特征工程施展不开。

而特征工程最能施展拳脚的地方就是工业界的异质表格(Tabulr)数据，同质数据是各列数据的单元要素接近，如文本对应字符，图片对应像素，语音对应语音frame。

这个结论在Yandex团队2021年论文 《**Revisiting Deep Learning Models for Tabular Data**》里印证了。这也是为什么早些年一票Kaggle比赛，清一色的XGB和LGB屠榜。
一言以蔽之的话，就是我之前文章的配图，**好风凭借力，助我上青云。**（写着写着怎么越来越像炒冷饭）

好风：在知识学习上恰当的模型。既不太复杂引入过多噪声和其他的问题，也没有太简单不足支撑，内在的机理有利于知识的学习。
借力：机器学习中使用特征工程的过程，人脑把数据经过处理，精炼，得到更接近结果的表达，更直白的可以得到预测目标。

特征工程的道与术

前面都在讲特征工程的道，而特征工程具体的术的话，其实也没有必要讲的特别详细了，但是我还是给你准备了一些关于术方面的资料的。
其实这部分真的没必要展开讲了，很多是熟能生巧，case by case，结合具体的业务的事情。

比如你们用的滑动验证码，这里面其实就有很多特征工程的东西，对鼠标的移动行为的各种角度的刻画，比如速度，加速度，角度等。
这里引用了知乎JovialCai的回答里的一张图。以风控场景为例，一些可能有用的数据如下（这里其实收集数据源的角度更大一些）：

我帮他拍几个很有用的特征
1.支付金额为整数的占比（刻画支付金额是不是都是整数）
2.支付金额分布前10的占比（刻画支付金额是不是集中在几个数里）
3.支付商铺的id占比（刻画支付金额是不是集中在几个店铺里）
4.非运营时段夜间交易行为数量（高危支付行为数量）

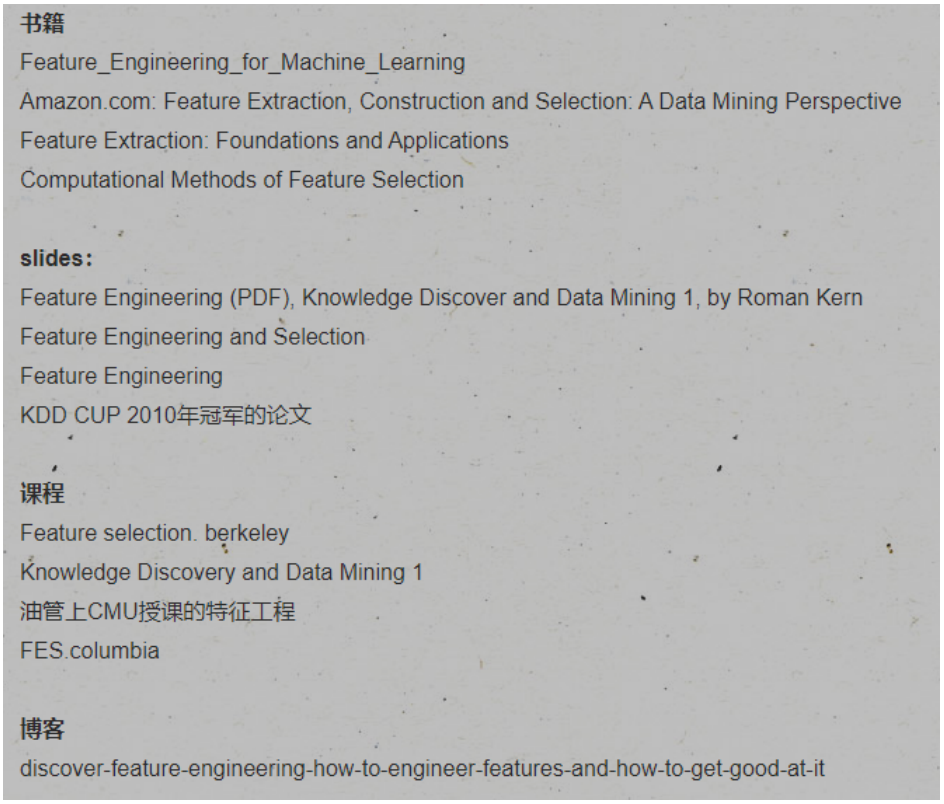
其他如图所示（引用了知乎JovialCai）：

数据源大类	原始数据字段	建议特征工程方向
支付流水	支付编号	日/周/月支付频率
	支付账户	来往账户数量、账户间关联图谱
	支付时间	最早最近支付时间、支付时段分布
	支付金额	支付金额总和/平均值/最大值
	支付地点	地点类型分布、较频繁地点
	支付目的	较频繁目的
	支付状态	支付成功/失败次数
财富管理	申购编号	申购频率
	申购时间	最早最近申购时间
	申购金额	申购金额总和/平均值/最大值
	产品类型	产品类型分布、产品偏好
	产品收益	收益总和、日均收益
	持仓金额	当前持仓、历史最大持仓、日均持仓
贷款信息	申请编号	申请频率
	申请时间	最早最近申请时间
	授信金额	授信金额总和/平均值/最大值
	提现金额	授信金额总和/平均值/最大值、提现比例
	资方类型	资方个数
	申请状态	申请通过/拒绝次数
	还款时间	提前结清/正常/逾期总天数
	逾期金额	逾期金额总和/最大值
	还款状态	已结清/正常/逾期笔数
app登录	登录编号	日/周/月登录频率
	登陆时间	最早最近登录时间、时段分布
	操作类型	操作类型分布、业务线偏好
电商流水	订单编号	当月/近3个月/近6个月/近12个月订单总数
	sku编号	当月/近3个月/近6个月/近12个月商品总数
	订单时间	最早最近订单时间、近12个月有消费月份数
	订单金额	当月/近3个月/近6个月/近12个月订单总金额/订单最大金额/平均单笔订单金额
	订单状态	当月/近3个月/近6个月/近12个月实付金额占比
	分期标识	当月/近3个月/近6个月/近12个月分期订单数占比
收货地址	订单编号	当月/近3个月/近6个月/近12个月使用收货地址个数
	订单时间	收货地址使用时长
	收货地址	城市等级、小区档次、地址稳定性、是否涉黑
	地址类型	最频繁收货地址类型、工作与住宅占比
运营商信息	通话数据	当月/近3个月/近6个月/近12个月通话量/通话次数、主叫/被叫/漫游通话量/通话次数占比、通话时段分布
	流量数据	当月/近3个月/近6个月/近12个月流量、流量时段分布
	账单信息	当月/近3个月/近6个月/近12个月账单金额平均值/最大值、当月储值金额、当前欠费金额
	客户信息	在网时长、在网状态、名下手机号码数量/终端设备数量/终端品牌
	互联网访问	各类别app访问总次数/总时长/活跃天数、app类别分布、是否非法网站

一些收集的术的资料

很多资料下载需要翻墙，整理的一些能下载下来的，公众号里回复“特征工程”即可获取。

附带链接地址（知乎上不好放外链）
https://www.zhihu.com/question/29316149/answer/607394337



书籍

Feature_Engineering_for_Machine_Learning

Amazon.com: Feature Extraction, Construction and Selection: A Data Mining Perspective

Feature Extraction: Foundations and Applications

Computational Methods of Feature Selection

slides:

Feature Engineering (PDF), Knowledge Discover and Data Mining 1, by Roman Kern

Feature Engineering and Selection

Feature Engineering

KDD CUP 2010年冠军的论文

课程

Feature selection. berkeley

Knowledge Discovery and Data Mining 1

油管上CMU授课的特征工程

FES.columbia

博客

discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it

“

历史精彩文章：

【闲谈】回看互联网十年校招薪资变化，我发现...

【段子】让算法工程师破防的瞬间

【经验】面试官带你破解算法岗诸神黄昏，神挡杀神！

【技术】一文串起从NLP到CV 预训练技术和范式演进

【经验】在读和转行进大厂做算法工程师的捷径

【技术】5行代码提升时间序列预测，都有用！

【闲谈】从Zillow用AI指导买房投资血亏说起

【段子】如何激怒一位算法工程师

【技术】5行代码实现的对比学习，效果超好！

【深度玄学】为何Bert三个Embedding可以相加

【技术】所有数据集上给神经网络刷分的通用方法

【闲谈】如何看顶会论文上关于泄露的乌龙

【经验】算法工程师的术与道：从特征工程谈数据敏感性

【技术】工业界文本分类避坑指南

【技术】从一道数学题面试题到GBDT原理的推导

【技术】Kaggle进阶：显著提分trick之指标优化

”

分享 收藏 点赞 在看

喜欢此内容的人还喜欢

对机器学习进行深入理解的一本好书
ChallengeHub

致初学者的深度学习入门系列（二）—— 卷积神经网络CNN基础
技术开发小圈

致初学者的深度学习入门系列（一）—— 简单神经网络基础
技术开发小圈

可能是全网特征工程实操最通透的...

原创 砍手豪 包大人 包包算法笔记 2022-02-14 09:00

收录于话题
#机器学习 10 #深度学习 11 #特征工程 2

点击蓝字关注我吧～

之前在我写的特征工程方法论里面提了一嘴，用automl搜索+人工启启发式可以高效稳定地完成特征工程，并取得不错的效果。
原文：可能是全网写特征工程最通透的...
不过这篇文章最大的问题是太过于侧重于**是什么，和为什么了**，至于**怎么办**埋了一个很大的伏笔。

包包算法笔记
包大人的算法，程序，机器学习，职场，理财闲谈。
62篇原创内容

公众号

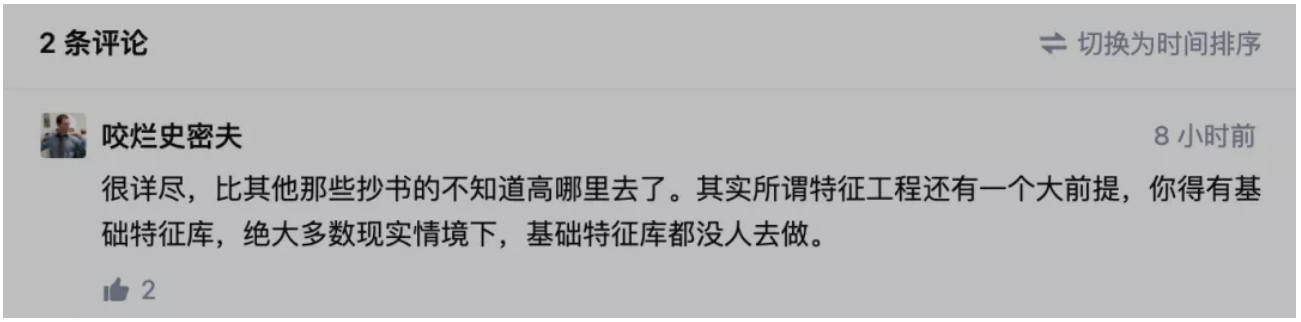
砍老师今天把这个思路给大家详细的写出了，可以实操。可以通过点击访问原文直达砍老师的知乎原文，作为kaggle历史总排名12的**GrandMaster**还有很多干货～

可能是全网特征工程**实操**最通透的...

背景

目前网上能搜到的讲特征工程方法基本都是教材里的那一套：缺失值填充，归一化，category特征one-hot，降维等等。但是指望靠这些提升模型性能是远远不够的，特别是对强大的xgb/lgb上述方法几乎是毫无意义。也有一些文章总结了特定业务的特征工程，但是对其他任务也没有泛化能力。

包大人插一嘴，这个评论很有水平，很多回答扯那些老掉牙的预处理。这篇文章基本就是从基础特征出发，衍生到高阶的实操方法论。这个评论的人可能是量化或者金融从业者，他们在基础因子库+启发式人力搜索上走了挺远了～



画导图抄书唬人，但是真的没什么水平～

砍手豪：本文探讨和介绍一下我的特征工程方法论：1.类automl的暴力特征字典思路 2.基于业务理解的特征工程思路 3.基于特征重要性的特征工程思路；然后是上述三者的反复迭代螺旋上升。

要点一

1.类automl的暴力词典搜索

暴力特征字典指的是当给定数据，能在想象力范围能组合出尽可能多的特征，并形成Pipline，加快特征尝试和迭代的速度，就像automl一样。打个比方，当给你两个类别特征A 和B，你能制造出多少个特征用于迭代？简单写十个：

```
count:A_COUNT、B_COUNT、A_B_COUNT
nunique: A_nunqiue_B （按B对称的下文省略）
ratio: A_B_COUNT/A_COUNT 在A里各个B类所占的比例
average:A_COUNT/A_nunqiue_B A里各个B类的平均数
most: A_most_B 在A类里出现最高的B是哪个
pivot: A_B1_count、A_B2_count A和B类里特定的B1、B2的联合统计
pivot2: A_B1_count-A_B2_count A的B1行为和B2行为的加减乘除
stat1: A_stat_A_B_COUNT 基于A_B_COUNT对A的描述，
stat2 : A_stat_B_COUNT 基于B_COUNT对A的描述，
序列化：初步LDA，NMF，SVD，进一步Word2Vec，doc2vec 再进一步 图神经网络deepwalk，pPRoNE
```

如果再加上numeric、time、target特征，几乎可以组合成无穷无尽的特征

提升方法：可以看各个数据挖掘的比赛获胜solution，我最初（17年）就是反复看当时几个kaggle GM plantgo&pipiu、Eureka&weiwei、Little Boat&jiwei liu的获胜方案开源，拓宽自己对特征工程的想象力。

缺陷：这类会产生大量特征，比如给五个category特征，就能组成(2**5-1)共31个count特征，自然也有大量无用特征，会降低模型质量和速度。

2.基于业务理解的特征工程思路

要点二

2.基于业务理解做特征

通过内在的业务逻辑去分做特征，可以先想业务逻辑，然后数据分析验证，也可以数据分析验证，然后得到业务逻辑，最大的好处是可解释性强，在此基础上泛化能力更强，而且模型规模小。举几个例子：

在Instacart Market Basket Analysis比赛，预测美国用户在线上商店的购物，我想我平时上班，买水果零食这种可买可不买的都放在周末，然后对这个数据里进行分析，发现在Instacart里，酒类商品的销量也集中在周末，因此做了很多item 和 time 交叉的特征，对模型提升较大。

在TalkingData AdTracking Fraud Detection Challenge比赛里，任务是判断虚假点击，通过数据分析发现低频IP容易是Fraud样本，仔细想这些Fraud点击都是自动化程序q每次随机生成的ip，因此容易是低频ip，而正常的ip因为是运营商动态分配共享的，因此普遍频率高。因此做了对channel，ad和ip频次的交叉特征，对模型提升较大。

提升方法：通过努力的数据分析，以及多交流获取业务的内在逻辑形式。在实际中就是多加几个行业群，多看论文多交流，在比赛中就是多逛论坛，看其他人的讨论。

缺陷：凭借业务逻辑做特征，容易遗漏掉强特征。很多时候并不能琢磨出全部的内在业务逻辑，甚至会主动的筛掉一些实际有价值的特征。

要点三

3.基于特征重要性表的特征工程思路

xgb/lgb可以输出特征重要性表，比起相关性分析，通过特征重要性表我们可以迅速在大量特征中获取强特征。在此基础上我们可以对强特征做更深层次的挖掘。

在Two sigma Rental-Listing-Inquireies里，GM little boat提到，既然manager id是强特征，那我们就可以用各种category，numeric特征去描述它。这里涉及到一个问题，很多人说FM，深度学习因为embedding的存在而具有了向新id泛化的能力，而树模型只会记忆。其实在我看来，特征工程就是一个人工embedding的过程，让高维度的类别特征数值向量化，因此也提升了树模型的泛化能力。回到这个比赛，就是特征重要性表为我们指明了特征工程努力的方向。在IJCAI2018 的比赛里，top2 solution 就是采用将特征重要性表靠前的数值特征暴力交叉，期望通过这种方法提高模型获取更多有价值的特征。

类似的，如果看到一个数值特征特征重要性很强，我们也可以用类别特征和其交叉。如果一个统计特征很重要，我们可以增加一个时区维度，比如最近一周，最近一个月的相应统计特征。如果距离上次时间很重要，我们可以增加距离上两次，上次三次的时间特征。等等。

进一步，特征重要性表也可以知道深度学习模型子结构的选择，序列特征对应rnn类，交叉特征q对应fm类，文本特征对应nlp类，如果特征不重要，就不用上相应的结构了，如果重要，就可以对将特定的特征输入对应的子结构了。

提升方法：经验的积累，如何将一个特征发散开来。

缺陷：首先得做出强特征，然后才能在强特征基础上发散，因此依赖一个好的特征重要性表

上文讲了三个我所认为最主要的特征工程思路，但是他们各有各的缺陷，因此
如何将其结合起来互补，螺旋迭代提升就是接下来能做的了。

要点四

4.类automl的暴力特征字典思路对基于业务理解的特征工程思路的协助

前文说到，基于业务理解的特征工程容易遗漏特征，不能挖掘全部可能存在的业务逻辑。那么我们可以先暴力特征字典全部罗列起来，然后在赋予其业务逻辑，看其在当前业务下是否有效。再回到第一项的暴力特征字典。我们把category A和B替换成user，item

```
count:user_COUNT（用户活跃度）、item_COUNT（商品热度）、user_item_COUNT（用户对特定商品的喜爱）
nunique: user_nunqiue_item  (一个用户购买多少种商品) item nunique_user (一个商品被多少个不同用户购买)
ratio: user_item_COUNT/user_COUNT （某个商品在user购买中的比例，喜爱程度）
average:user_COUNT/user_nunqiue_item （平均每类商品的购买量）
most: user_most_item （用户最喜爱的品类）
pivot: user_item1_count、user_item2_count （用户和特定商品的交互）
pivot2: user_item1_count-user_item2_count （用户不同行为的差值，比如生活用品和娱乐用品的比例）
stat1: user_stat_user_item_COUNT （max:买的最多的商品的数量，std：不同商品的分散度，是专宠还是偏爱）
stat2 : user_stat_item_COUNT （mean:用户是喜欢热门商品还是冷门商品）
序列化：初步LDA，NMF，SVD（用商品描述用户画像）
进一步Word2Vec，doc2vec 再进一步 图神经网络deepwalk，pPRoNE（刻画商品和用户的共现性和相似性）
```

要点五

5.类automl的暴力特征字典思路对基于特征重要性表的特征工程思路的协助

首先我们原始 data去跑特征重要性表，知道某个category特征或numeric很重要，要进一步挖掘这个特征的时候，比如前文说的“在Two sigma Rental-Listing-Inquireies里，GM little boat提到，既然manager id是强特征，那我们就可以用各种category，numeric特征去描述它。”我们就可以基于暴力特征字典去强化这个特征，看看如何去拓展这个强特征的维度。

要点六

6.基于业务理解的特征工程思路 和 基于特征重要性表的特征工程思路 对 类automl的暴力特征字典思路的协助

类automl的暴力特征字典思路最大的问题是可以产生无数的特征，比如五个类别特征就能产生31种count特征，这时候我们可以基于特征重要性表，把特征重要性低的类别特征从组合中删去，也可以基于业务理解，把一些明显无相关性的category交叉移除。这样就不会产生过多无用的特征变成噪音降低模型速度和精度。

要点七

7.基于业务理解的特征工程思路 和 基于特征重要性表的特征工程思路 的相互迭代

其实比起数据分析，特征重要性表是一个可以更快的理解业务逻辑的方法
如果一个特征重要性表里存在一个我们原本认为应该无关紧要的特征却有很高的重要性，其实就可以增强我们对业务的理解，我们需要从业务角度思考为什么这个特征有好的效果，然后从业务角度上去做一个更好的特征。
比如特征重要性表里category A 和 numeric B特征都很重要，虽然无论树模型还是深度学习模型都已经有很强的特征交叉能力了，但经过业务分析，其实是 A_mean_B特征影响结果，原始的A和B还是不如我们直接把A_mean_B做出来效果好。

因此，通过观察特征重要性表，思考背后真正的业务逻辑，找出真正和target直接相关的特征，既能提升对业务的理解，也能够提升模型的性能。

在Avito Demand Prediction Challenge（类似闲鱼的一个app转化预测）里，大家发现各种**category_mean_price - price**有很高的特征重要性，因此冠军little boat思考出这不就是合理价格和卖家出价的差影响转化率嘛，于是干脆先建了一个子模型，先预测出pred_price,然后用pred_price-price用于转化率模型，取得了更好的效果，这就是基于特征重要性来理解业务，深挖特征的一个好的案例。

总之，就是三种特征工程思路相互补充，反复迭代，最后通过验证集取得一个好的特征组合。

小结

本文从方法论角度探讨和总结了我的特征工程方法，基本上毫无保留。但是要反思的就是，使用这一套方法论是无法和最好的特征工程大师（比如国内的江离、otto数据挖掘俱乐部）还是有很大差距，我猜测一下可能用以下两种原因：

1. 高手们还有其他角度的特征工程构造逻辑
2. 现有的特征工程逻辑我做的还不够好，比如即便我观察特征重要性表知道某些特征很重要后，也经常无法真正挖掘出反应业务逻辑的深层特征，需要后续看其他人的开源才能焕然大悟。

在这里抛砖引玉，供大家参考。
