数据挖掘小组报告分享

自动化特征工程与建模的尝试

周千鸿、陈靖洋

复旦大学管理学院

2021 年 12 月 24 日

研究背景

研究背景

- 一学期的机器学习课中，我们接触了各种模型和数据集
- 然而每遇到一个新的数据集都要重新进行特征工程、调参，重复工作大量消耗我们的时间。
  - 是否存在流程抽象化的工具？
  - 部分流程中是否存在更高效的方法？

主要思想和目的

- 创建自动化特征工程与建模 pipeline 工具，不用基于 Domain Knowledge 快速易操作对表格数据集进行挖掘。
- 实现方式：流程自动化与算法自动化

研究背景

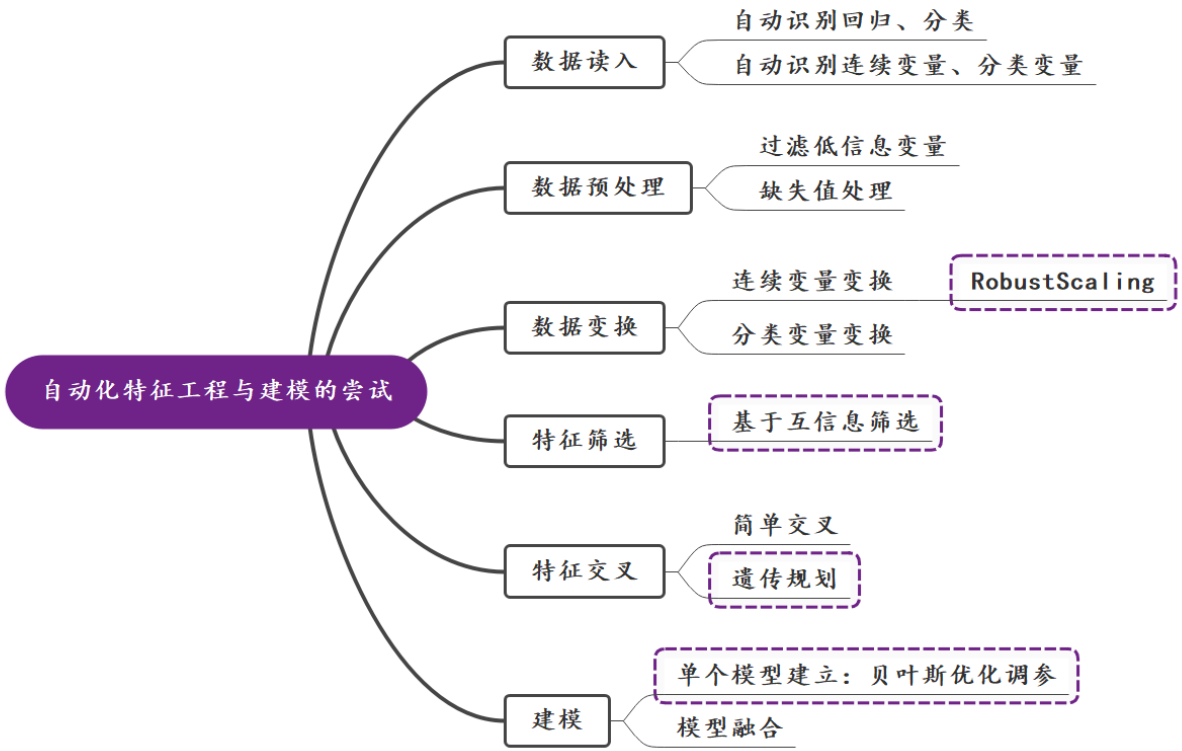
研究背景

- 算法自动化
  - 自动化特征工程
  - 自动化模型调参
- 流程自动化

```
freemindpipeline = Pipeline(
[
        ('preprocess',preprocess_pipeline),
        ('datatransform',DataTransformer()),
        ('cross',featurecross),
        ('genetic_programming',GpGenerate(enable=enable_gp)),
        ('modeling',Modeler(use_raw_data=True,class_balance=class_balance))
    ]
)
freemindpipeline.fit(train_data)
predict_results = freemindpipeline.predict(test_data)
```

研究框架

Pipeline 建模方法



Stage 1：数据读入与预处理

数据读入：读取数据并判断工作任务和变量属性

- 根据因变量自动识别回归、分类问题
- 自动识别连续变量、分类变量

Stage 1：数据读入与预处理

数据预处理：基本数据预处理

- 过滤低信息变量
  - 删除方差/变异系数过小的连续变量
  - 删除水平过多或单个水平占大多数的分类变量
- 缺失值处理
  - 分类变量：缺失值自成一类
  - 连续变量：
    - 缺失比例较小 $\rightarrow$ 中位数插补
    - 缺失比例较大 $\rightarrow$ 创建 Indicator 并删除原变量
    - 剩余缺失变量迭代插补（有监督）

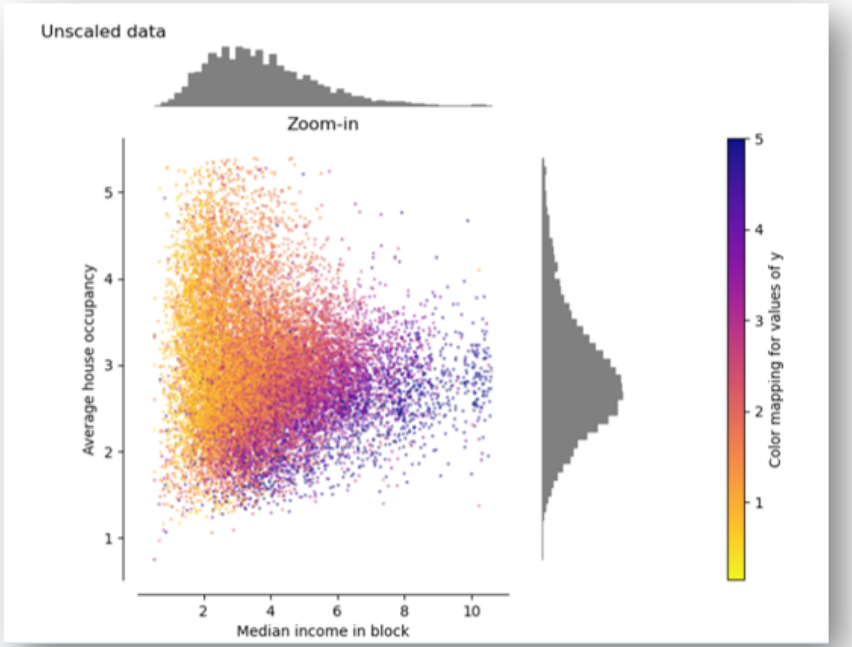
Stage 2: 数据变换

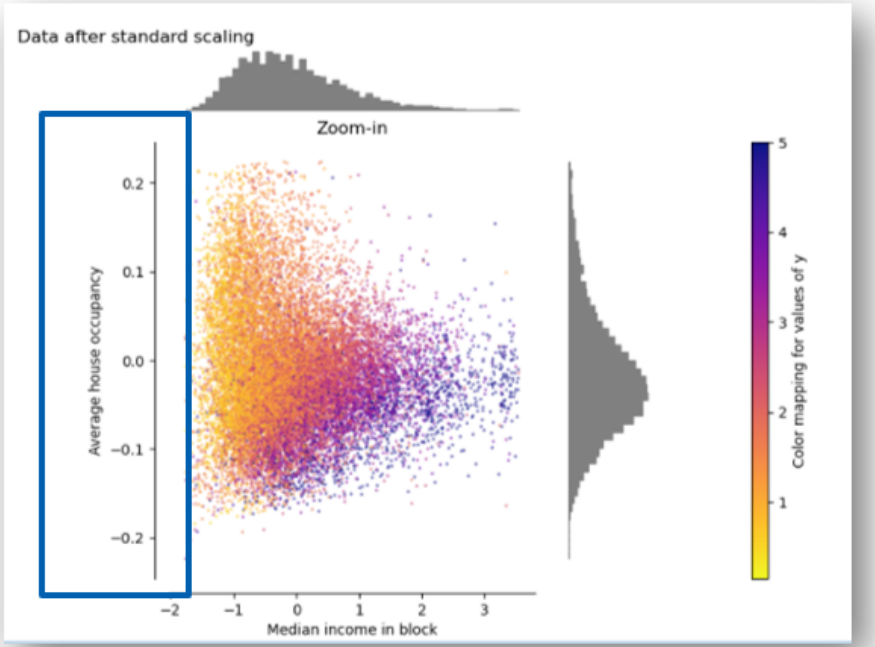
连续变量变换: 在不受异常值的影响下消除量纲影响

- 常规方法：标准化处理 Standard Scaling 方法
  用均值和标准差缩放数据
  - 变换方式：$X_i^* = \frac{X_i - mean(X)}{Var(X)}$
  - 问题：容易受到异常值影响，方差不够稳健
- 改进方法：区间缩放 Robust Scaling 方法
  用中位数和截尾方差缩放数据
  - 变换方式: $X_i^* = \frac{X_i - median(X)}{IQR(X)}$
    - IQR 是第 1 个四分位数和第 3 个四分位数 (第 75 分位数) 之间的范围。
  - 相对优势：增加对于异常值的稳健性

Stage 2: 数据变换

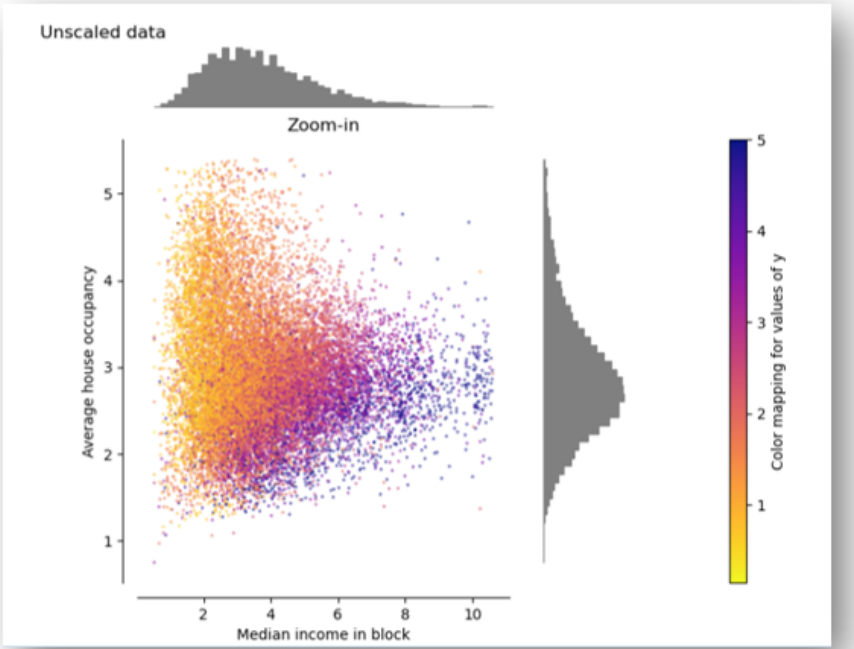
连续数据处理-Standard Scaling 方法

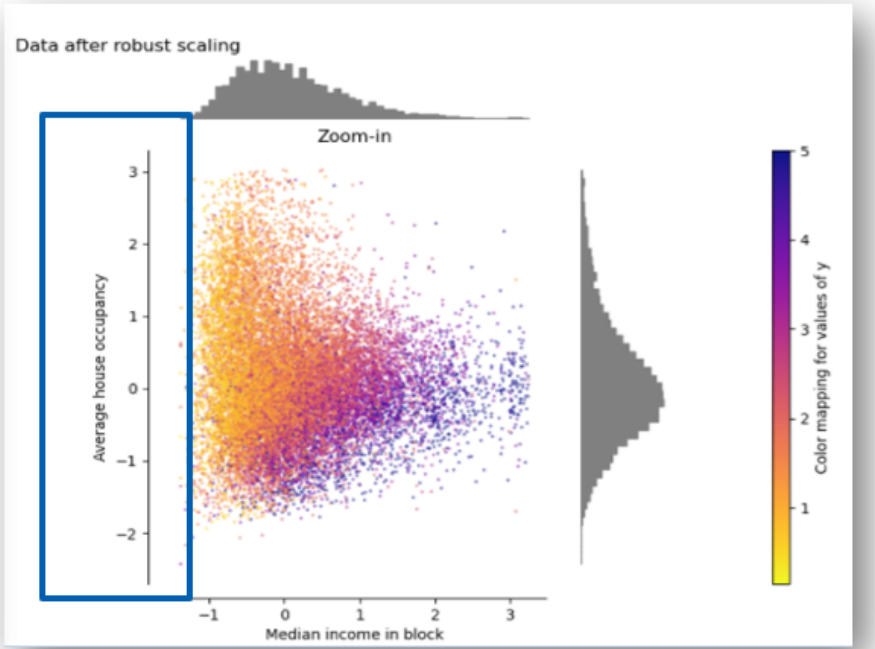




Stage 2: 数据变换

连续数据处理-Robust Scaling 方法





Stage 2: 数据变换

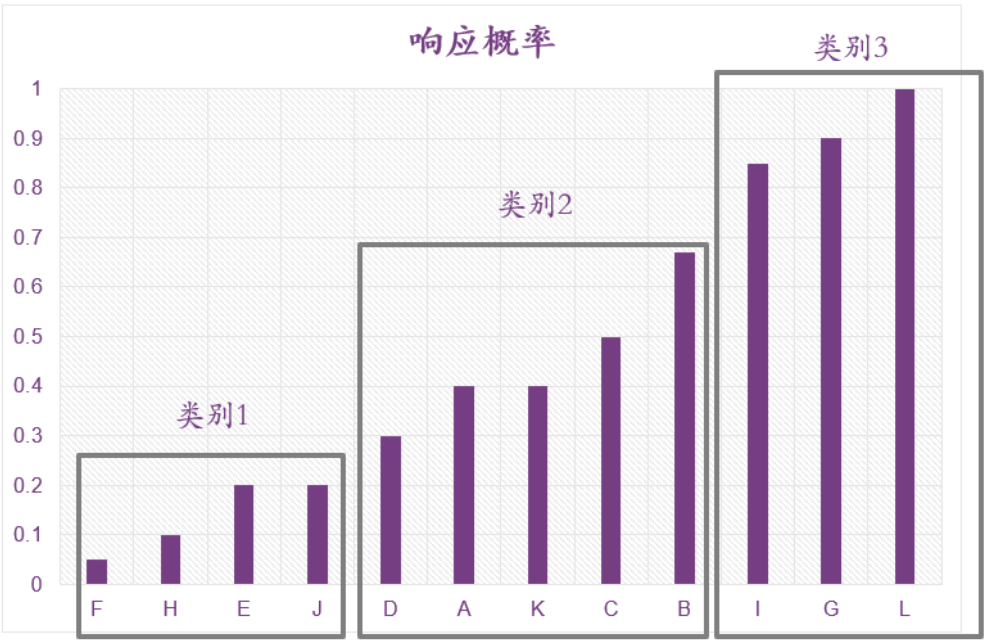
分类变量变换: 对分类变量水平进行聚类

- 处理动机：如果变量水平本身较多，那么哑变量的水平个数也会相应变多，对特征工程造成负面影响，需要将分类变量的水平进行压缩处理。
- 处理方法：变量水平聚类

Stage 2: 数据变换

分类数据处理-变量水平聚类

feature	target
A	0
A	0
A	0
A	1
A	1
B	1
B	1
B	0
C	1
C	0
...	
...	
L	1
L	1
L	1



Stage 3：特征筛选

- 常用方法：相关系数方法
  - 通过 Pearson/Spearman 等相关系数进行变量筛选
  - 但会出现失效情况：变量之间不是线性关系
- 方法改进：基于互信息进行变量筛选

$$I(X,Y) = H(X) - H(X|Y)$$

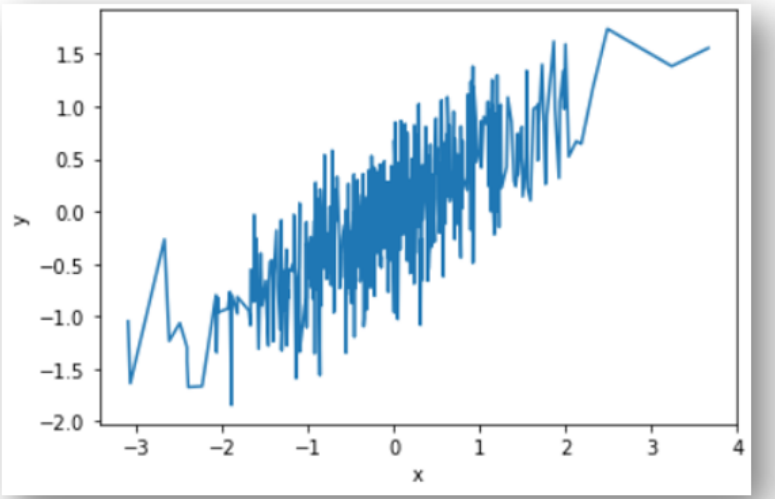
  - H 表示信息熵。I(X,Y) 表示知道 X 的信息后，Y 的不确定性减少了多少，如减少较多，则特征包含信息越多。
  - 互信息也可以表示成联合密度和边际密度乘积的 KL 散度，基于概率密度算出的 feature dependence，具有很高的可靠度。以上的公式一般适用于两个变量都是分类变量，如果有连续变量可以用 K 近邻来实现[1]

$$I(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)log\frac{p(x,y)}{p(x)p(y)}$$

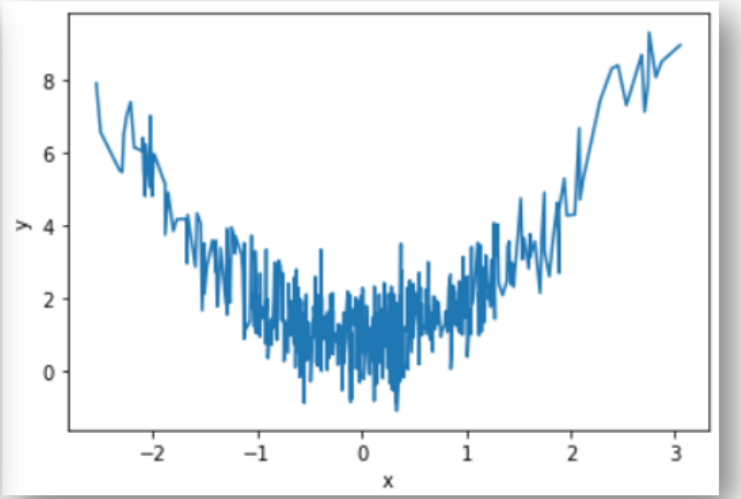
[1] B. C. Ross "Mutual Information between Discrete and Continuous Data Sets". PLoS ONE 9(2), 2014.

Stage 3: 特征筛选

相关系数失效情况



$y = 0.5x+1$
Pearson : 0.7786
互信息 : 0.51749881



$y = x^2+1$
Pearson : 0.08343
互信息 : 0.66063355

Stage 4：特征交叉

简单交叉：对分类变量新造有效特征

- 分类变量与分类变量：笛卡尔乘积生成特征
- 分类变量与连续变量：通过分组均值和分组方差生成两列特征

遗传规划：对连续变量新造有效特征

- 一种机器自动挖因子（特征）的有监督学习算法，基于遗传算法。随机生成公式树，公式树如下图所示，适应度高的公式树生存下来，通过突变，基因重组、杂交产生下一代公式树，进入下一次迭代。
- 适应度：采用互信息。
- 最终迭代完成后选出一定数目的公式树，作为新的因子。

Stage 4：特征交叉

简单交叉：对分类变量新造有效特征

离散和离散变量特征交叉

特征1
1
2
3

特征2
A
B
C

新特征
1A
1B
1C
2A
2B
2C
3A
3B
3C

编码
1
2
3
4
5
6
7
8
9

连续和离散变量特征交叉

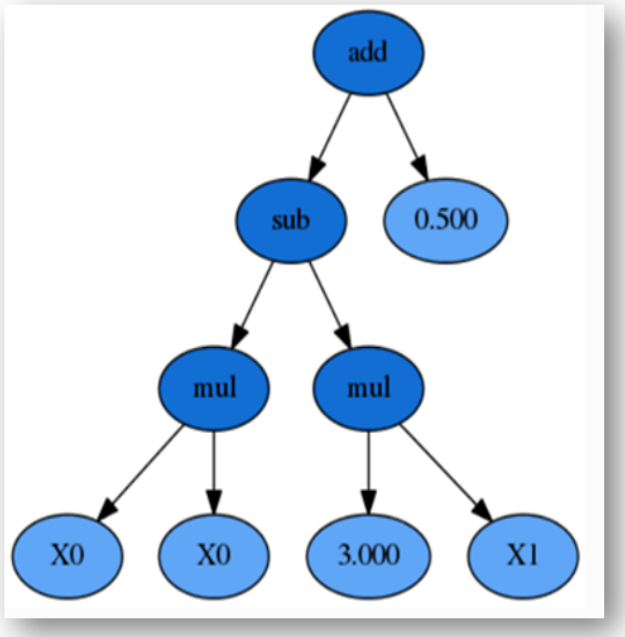
离散特征
A
A
B

连续特征
12
19
31

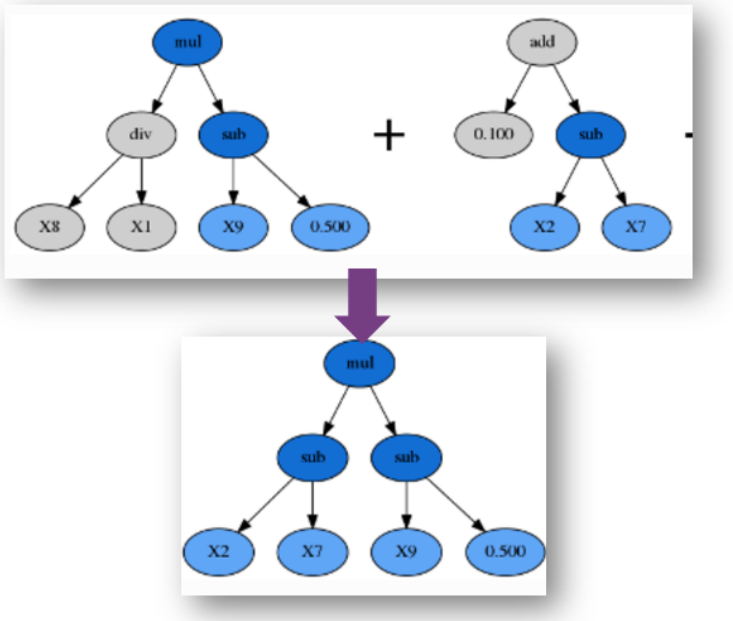
离散特征	分组均值	分组标准差
A	11.75	3.30
B	3.667	1.5275
C	27.75	2.754
D	25.667	24.007

Stage 4: 特征交叉

遗传规划：对连续变量新造有效特征



一个交叉变换的实例：



Stage 5：模型建立

单一模型：针对所选特征建立模型

- 寻找超参数是优化问题，由于无梯度，评价一次非常花时间。
- 常见的参数搜索方法，如网格搜索和随机搜索都是互相独立的，下一次搜索不会使用过去搜索的信息，所以效率比较低。
- 贝叶斯优化就是，利用之前的搜索信息建立高斯过程，得到 Y 的后验分布，并通过收益函数权衡探索与利用，找到最合适的点去搜索。

模型融合：通过 XGBT/LGBM 进行模型融合

- 划出一个 holdout set，使用贝叶斯优化分别调出最好的 XGB/LGBM 模型。并使用 stacking 将两个模型进行融合，得到最终模型。

Stage 5：模型建立

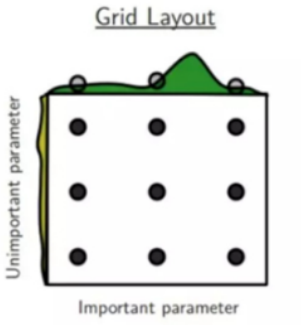
参数搜索：贝叶斯优化

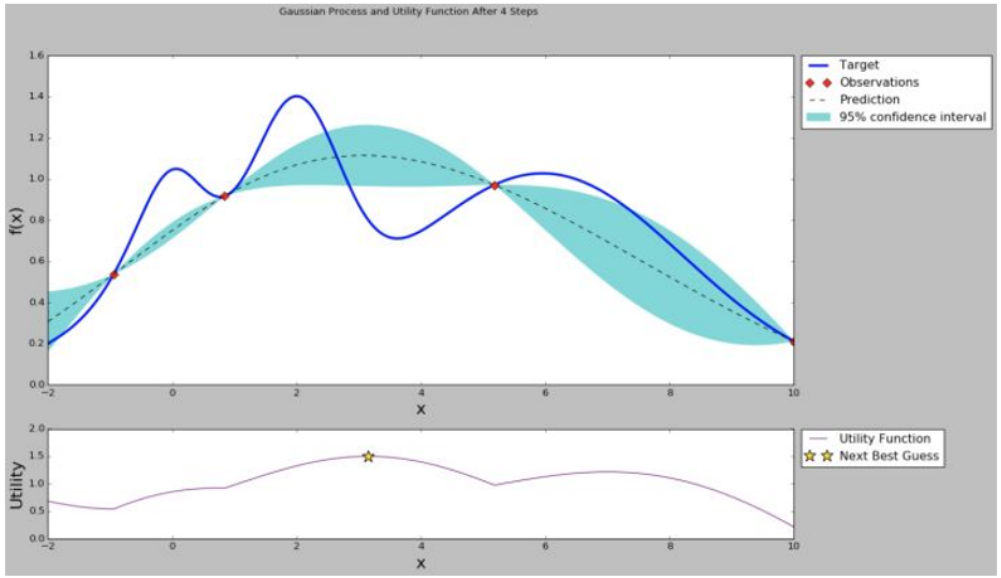
Algorithm 1 Sequential Model-Based Optimization

Input: $f, \mathcal{X}, S, \mathcal{M}$
$\mathcal{D} \leftarrow \text{INITSAMPLES}(f, \mathcal{X})$
for $i \leftarrow |\mathcal{D}|$ **to** T **do**
  $p(y \mid \mathbf{x}, \mathcal{D}) \leftarrow \text{FITMODEL}(\mathcal{M}, \mathcal{D})$
  $\mathbf{x}_i \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}} S(\mathbf{x}, p(y \mid \mathbf{x}, \mathcal{D}))$
  $y_i \leftarrow f(\mathbf{x}_i)$ ▷ Expensive step
  $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{x}_i, y_i)$
end for

Stage 5：模型建立

参数搜索：贝叶斯优化





案例分析 1：使用 5 个数据集来验证 pipeline 的效果

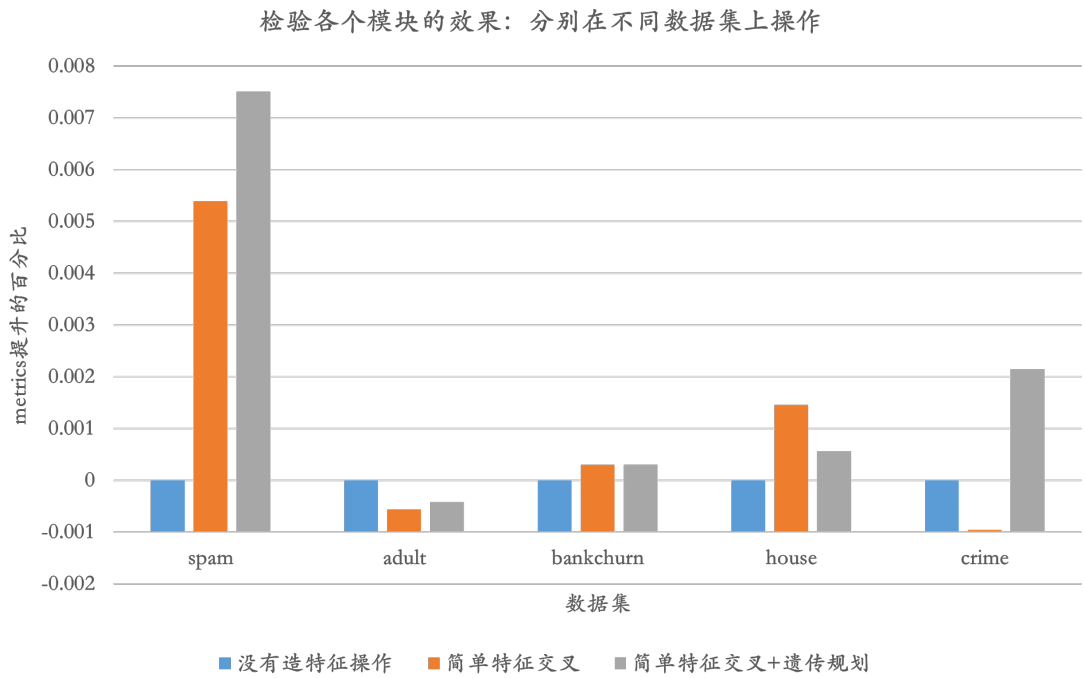
数据集[2]

数据集	数据描述	训练数量	特征	输出	**measure**
spam	垃圾邮件识别	4601	57	分类	f-score
adult	成年人是否高收入识别	30162	14	分类	f-score
bankchurn	信贷违约识别	7000	19	分类	f-score
house	房价预测	1100	79	连续	mape
crime	犯罪率预测	1560	127	连续	mape

[2] https://archive.ics.uci.edu/ml/datasets.php, https://www.kaggle.com/datasets

案例分析 1：使用 5 个数据集来验证 pipeline 的效果

- 为了验证不同模块效果，分别做几次实验：不做特征交叉、只做简单特征交叉、做简单特征交叉与遗传规划，建模步骤都相同。记录他们在测试集上的 metrics 的提升百分比（连续变量使用 mape 作为 metrics，离散变量使用 F-score）



案例分析 2：天池数据集-信用卡违约预测[3]

- 信用卡违约预测，二分类问题
- 训练样本 80 万，测试 20 万，使用 AUC 为评价准则
- 使用十行代码调用 pipeline，在没有调参的情况下 AUC 为 0.7350，（比赛第一名 AUC 为 0.7497），相差不算太大

```
def create_pipeline(enable_cross=True,enable_gp=True,class_balance=True):
    freemindpipeline = Pipeline(
    [
            ('preprocess',preprocess_pipeline),
            ('datatransform',DataTransformer()),
            ('cross',featurecross),
            ('genetic_programming',GpGenerate(enable=enable_gp)),
            ('modeling',Modeler(use_raw_data=True,class_balance=class_balance))
        ]
    )
    return freemindpipeline
```

```
x_train,y_train,x_test,y_test = get_large_bankchurn_data()
todatawrapper = toDataWrapper()
train_wrapper = todatawrapper.convert(x_train,y_train)
test_wrapper = todatawrapper.convert(x_test)
freemindpipeline = create_pipeline(enable_cross=True,enable_gp=True,class_balance=False)
freemindpipeline.fit(train_wrapper)
y_test_predict = freemindpipeline.predict_proba(test_wrapper)
pd.DataFrame(y_test_predict).to_csv("bankchurn_predict.csv",index=False)
```

[3] https://tianchi.aliyun.com/competition/entrance/531830/information

- 整个 pipeline 其实存在很多超参数，比如变量水平聚类聚类的目标类个数、互信息筛选变量的阈值、遗传规划中保留的因子数目等等。想要做出更精细的结果还是依赖于参数的调整。没有能够做到参数选取的自动化。
- 自动化是一个听起来很美好但是做起来非常困难的操作。只依靠不引入任何先验知识的挖掘带来提升是不容易的。Domain knowledge 依然不可或缺，懂业务的 Data Scientist 有着重要影响。
- 感谢刘老师这学期以来的悉心教导，您辛苦了！

Thanks!