因子筛选与特征工程调研报告

因子筛选与特征工程一般分为基于单因子在模型中效果检验的筛选与基于对模型直接目标效果的特征选择

单因子测试

回归法

回归法是检验因子有效性最常用的方法之一。具体方法是对T时期的因子暴露向量与T+1时期的股票收益向量进行线性回归，得到的回归系数为该因子在该时期的因子收益，其显着性水平本期回归中的因子回报（t值）。个股在某一横截面的因子敞口是指个股在当前时间对该因子的因子值。 T时期回归模型的具体表达式如下：

$$r^{T+1} = X^T a^T + \sum_j \text{Indus}_j^T b_j^T + \ln\_\text{mkt}^T b^T + \varepsilon^T$$

其中

- r^{T+1}: 所有股票在T+1时期的收益
- X^T: T时期所有股票在测试单因子上的暴露向量
- $Indus_j^T$: T时期所有个股对j行业因子的暴露向量（0/1哑变量）
- $\ln\_mkt$: 基于对数市值因子的T期间所有股票的风险敞口向量
- a^T, b^T, b_j^T: 对应因子，要拟合的常数，往往更关注 a^T

评估：

1. t值序列绝对值的平均值：因子重要性的重要标准
2. t值序列绝对值大于2的比例：判断因子的显著性是否稳定
3. t值序列的平均值：与1.结合。可以判断t值的正负方向是否稳定的因素
4. 因子收益率序列的平均值：确定因子收益率的大小。

IC 分析

该因子的IC值是指该因子在T期的敞口向量与T+1期的股票收益向量之间的相关系数，即：

$$IC^T = \text{corr}\left(r^{T+1}, X^T\right)$$

在上述公式中，因子暴露向量X^T通常不直接使用原始因子值，而是通过去极值和中性化等方法。 在实际计算中，Pearson相关系数的使用可能会受到因子极值的很大影响，而Spearman秩相关系数的使用更稳健。以这种方式计算的IC通常被称为Rank IC。IC分析模型构建如下：

1. 股票池、回溯区间和截面期与回归方法相同
2. 首先对因子暴露向量进行一定的预处理，然后计算处理后的T周期因子暴露向量与T+1周期股票收益向量之间的Spearman相关系数，作为T周期因子Rank IC值

评估：

- Rank IC值序列平均值：因子显著性
- Rank IC值系列标准差：因子稳定性
- ICIR (Rank IC值序列均值与标准差之比)：因子效度
- Rank IC值系列大于零的比例：因子的作用方向是否稳定

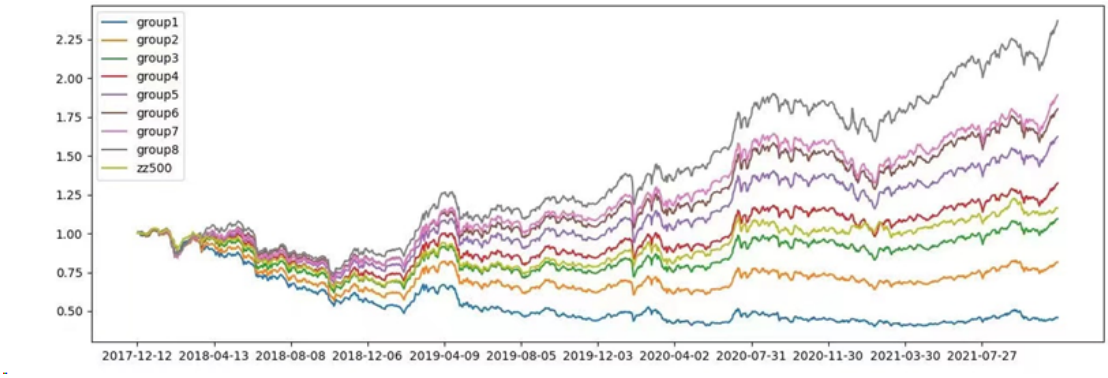
分层回测（主要）

依照因子值对股票进行打分，构建投资组合回测，是最直观的衡量因子优劣的手段。分层测试法与回归法、IC值分析相比，能够发掘因子对收益预测的非线性规律。也即，若存在一个因子分层测试结果显示，其Top组和Bottom组的绩效长期稳定地差于Middle组，则该因子对收益预测存在稳定的非线性规律，但在回归法和IC值分析过程中很可能被判定为无效因子。分层测试模型构建方法如下：

1. 股票池、回溯区间、截面期均与回归法相同
2. 换仓：在每个截面期核算因子值，构建分层组合，在截面期下一个交易日按当日收盘价换仓
3. 分层方法：先将因子暴露度向量进行一定预处理，将股票池内所有个股按处理后的因子值从大到小进行排序，等分N层，每层内部的个股等权重配置。当个股总数目无法被N整除时采用任一种近似方法处理均可，实际上对分层组合的回测结果影响很小。分层测试中的基准组合为股票池内所有股票的等权组合
4. 多空组合收益计算方法：用Top组每天的收益减去Bottom组每天的收益，得到每日多空收益序列 r_n ，则多空组合在第n天的净值等于 $(1+r_1)(1+r_2)...(1+r_n)$

评估：

全部N层组合年化收益率（观察是否单调变化），多空组合的年化收益率、Sharpe



因子预处理

中位数去极值

设第T期某因子在所有个股上的暴露度向量为 D_i，D_M 为该向量中位数，D_{M1} 为向量$|D_i - D_M|$的中位数，则将所有大于 $D_M + 5D_{M1}$ 的数重设为 $D_M + 5D_{M1}$，将向量 D_i 中所有小于 $D_M - 5D_{M1}$ 的数重设为$D_M - 5D_{M1}$

中性化

以行业及市值中性化为例，在第T期截面上用因子值（已去极值）做因变量、对数总市值因子（已去极值）及全部行业因子（0/1哑变量）做自变量进行线性回归，取残差作为因子值的一个替代，这样做可以消除行业和市值因素对因子的影响

标准化

将经过以上处理后的因子暴露度序列减去其现在的均值、除以其标准差，得到一个新的近似服从 $N(0,1)$ 分布的序列，这样做可以让不同因子的暴露度之间具有可比性

缺失值处理

单因子测试，为了不干扰测试结果，一般不填补缺失值，在构建完整多因子模型时需考虑填补缺失值

特征选择

在处理结构型数据时，特征工程中的特征选择是很重要的一个环节，特征选择是选择对模型重要的特征。它的好处在于:

- 减少训练数据大小，加快模型训练速度
- 减少模型复杂度，避免过拟合
- 特征数少，有利于解释模型
- 如果选择对的特征子集，模型准确率可能会提升

具体地，通常可以:

- 去除冗余无用特征，减低模型学习难度，减少数据噪声
- 去除标注性强的特征，例如某些特征在训练集和测试集分布严重不一致，去除他们有利于避免过拟合
- 选用不同特征子集去预测不同的目标

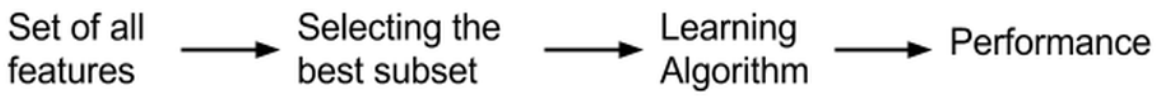
三大类方法

根据特征选择的形式，可分为三大类:

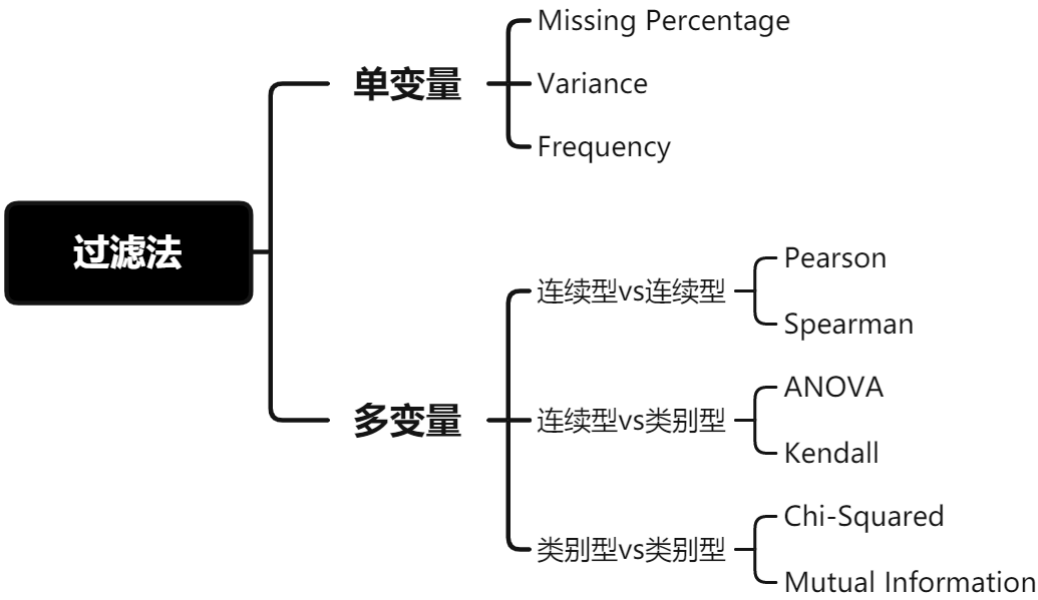
- Filter(过滤法)：按照 `发散性` 或 `相关性` 对各个特征进行评分，设定阈值或者待选择特征的个数进行筛选
- Wrapper(包装法)：根据目标函数（往往是预测效果评分），每次选择若干特征，或者排除若干特征
- Embedded(嵌入法)：先使用某些机器学习的模型进行训练，得到各个特征的权值系数，根据系数从大到小选择特征（类似于Filter，只不过系数是通过训练得来的）

过滤法

过滤法总流程如下:



具体分类如下:



单变量

(1) 缺失百分比(Missing Percentage)

缺失样本比例过多且难以填补的特征，建议剔除该变量。

(2) 方差(Variance)

若某连续型变量的方差接近于0，说明其特征值趋向于单一值的状态，对模型帮助不大，建议剔除该变量。

(3) 频数(Frequency)

若某类别型变量的枚举值样本量占比分布，集中在单一某枚举值上，建议剔除该变量。

多变量

研究多变量之间的关系时，主要从两种关系出发:

- **自变量与自变量之间的相关性**: 相关性越高，会引发**多重共线性**问题，进而导致模型稳定性变差，样本微小扰动都会带来大的参数变化[5]，建议在具有共线性的特征中选择一个即可，其余剔除。
- **自变量和因变量之间的相关性**: 相关性越高，说明特征对模型预测目标更重要，建议保留。

由于变量分连续型变量和类别型变量，所以在研究变量间关系时，也要选用不同的方法:

连续型vs连续型

主要有:

- 皮尔逊相关系数(Pearson Correlation Coefficient)，也即IC
- 斯皮尔曼相关系数(Spearman's Rank Correlation Coefficient)，也即RankIC

连续型vs类别型

主要有:

方差分析(Analysis of variance, ANOVA)

肯德尔等级相关系数(Kendall tau rank correlation coefficient)

Kendall系数计算公式：

$$p = \frac{N_{\text{Concordant Pairs}} - N_{\text{Discordant Pairs}}}{\frac{n \times (n-1)}{2}}$$

其中 $N_{\text{Concordant Pairs}}$ 是同序对， $N_{\text{Discordant Pairs}}$ 是异序对，该系数反映了同序对在总对数中的占比。

类别型vs类别型

卡方检验(Chi-squared Test)

互信息(Mutual Information)

互信息是**衡量变量之间相互依赖程度**，它的计算公式如下：

$$\begin{aligned}
I(X;Y) &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\
&= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)} - \sum_{x,y} p(x,y) \log p(y) \\
&= \sum_{x,y} p(x)p(y \mid x) \log p(y \mid x) - \sum_{x,y} p(x,y) \log p(y) \\
&= \sum_{x} p(x) \left(\sum_{y} p(y \mid x) \log p(y \mid x) \right) - \sum_{y} \log p(y) \left(\sum_{x} p(x,y) \right) \\
&= -\sum_{x} p(x) H(Y \mid X = x) - \sum_{y} \log p(y)p(y) \\
&= -H(Y \mid X) + H(Y) \\
&= H(Y) - H(Y \mid X)
\end{aligned}$$

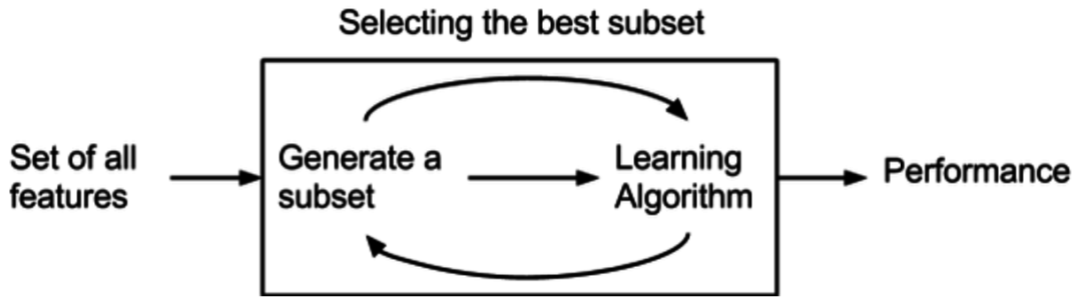
它可以转为熵的表现形式，其中 $H(H|Y)$ 和 $H(Y|X)$ 是条件熵， $H(H,Y)$ 是联合熵。当 X 与 Y 独立时， $p(x,y) = p(x)p(y)$ ，则互信息为0。当两个变量完全相同时，互信息最大，因此互信息越大，变量相关性越强。此外，互信息是正数且具有对称性(即 $I(X;Y) = I(Y;X)$)。

总结

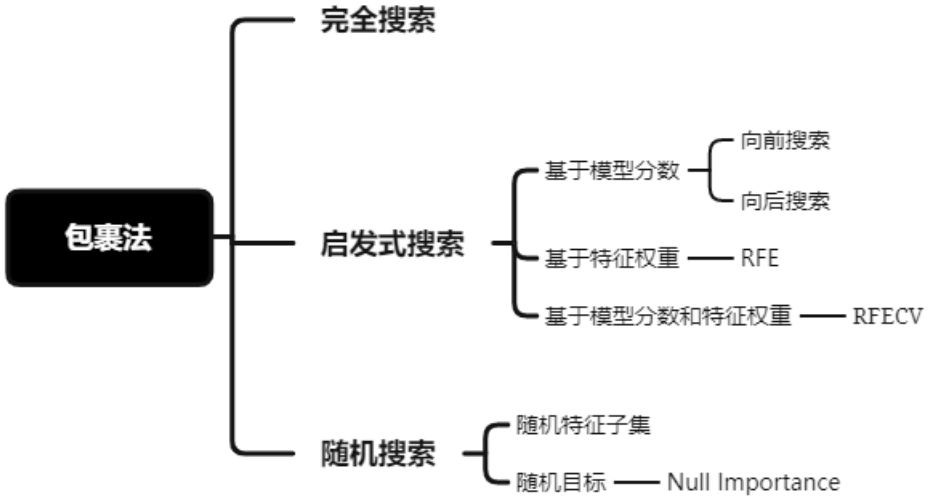
总结以上内容，可以得到下表

变量	适用特征	计算	规则	仅线性	备注	python调用
皮尔逊相关系数(Pearson Correlation Coefficient)	多变量:连续型vs连续型	$\rho(X,Y)$	剔除相关系数接近或等于0的特征	是	样本需符合正态分布	sklearn.feature_selection.f_regression
斯皮尔曼相关系数(Spearman's Rank Correlation Coefficient)	多变量:连续型vs连续型	$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n\left(n^2-1\right)}$	剔除相关系数接近或等于0的特征	否	也适合有序类别。不对样本分布做要求	scipy.stats.spearmanr
方差分析(Analysis of variance,ANOVA)	多变量:连续型vs类别型	$F = \frac{MSB}{MSE}$	剔除F值过低的特征,或者剔除p值<0.05的特征	是	不要求类别有序,但总体样本要具备方差同质性、要求组内样本服从正态分布、样本独立	sklearn.feature_selection.f_classif
肯德尔等级相关系数(Kendall tau rank correlation coefficient)	多变量:连续型vs类别型	$p = \frac{N_{\text{Concordant Pairs}} - N_{\text{Discordant Pairs}}}{\frac{n \times (n-1)}{2}}$	剔除相关系数接近或等于0的特征	否	要求类别有序	scipy.stats.kendalltau
卡方检验(Chi-squared Test)	多变量:类别型vs类别型	$\chi 2 = \sum \frac{(observed - expected)^2}{expected}$	剔除卡方值过低的特征,或者剔除p值<0.05的特征	否	不要求类别有序	sklearn.feature_selection.chi2
互信息(Mutual Information)	多变量:类别型vs类别型	$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$	剔除互信息接近或等于0的特征	否	不要求类别有序,也可用于连续型变量间的相关性分析,不能用于稀疏矩阵	sklearn.feature_selection.mutual_info_classif/mutual_info_regression

包装法



具体分类如下:



完全搜索

遍历所有可能组合的特征子集，然后输入模型，选择最佳模型分数的特征子集（不推荐，开销过大）

启发式搜索

启发式搜索是利用启发式信息不断缩小搜索空间的方法。在特征选择中，模型分数或特征权重可作为启发式信息

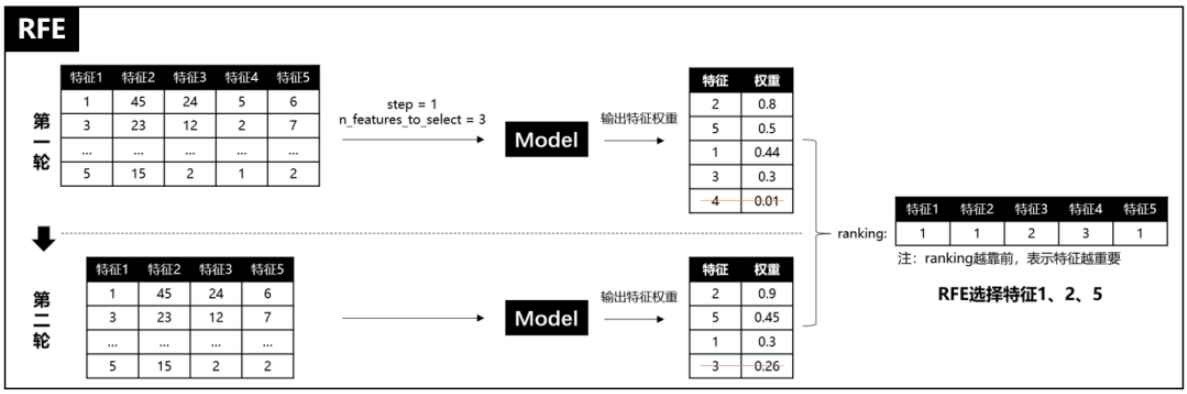
向前/向后搜索

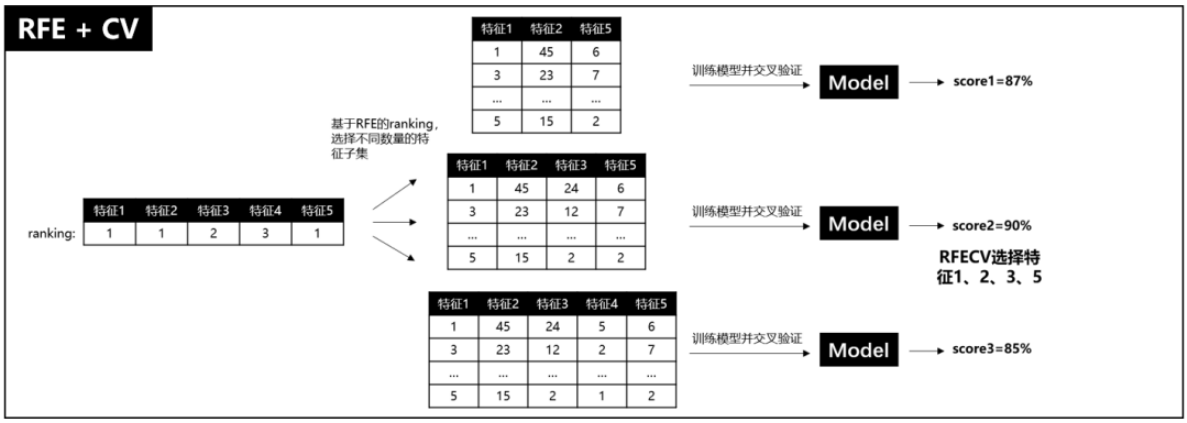
向前搜索是先从空集开始，每轮只加入一个特征，然后训练模型，若模型评估分数提高，则保留该轮加入的特征，否则丢弃。反之，向后特征是做减法，先从全特征集开始，每轮减去一个特征，若模型表现减低，则保留特征，否则弃之。

递归特征消除

递归特征消除简称**RFE(Recursive Feature Elimination)**，RFE是使用一个基模型进行多轮训练，每轮训练后，消除若干低权值(例特征权重系数或者特征重要性)的特征，再基于新的特征集进行下一轮训练。

RFE使用时，要提前限定最后选择的特征数`(n_features_to_select)`，这个超参很难保证一次就设置合理，因为设高了，容易特征冗余，设低了，可能会过滤掉相对重要的特征。而且RFE只是单纯基于特征权重去选择，没有考虑模型表现，因此RFECV出现了，REFCV是REF + CV，它的运行机制是：先使用REF获取各个特征的ranking，然后再基于ranking，依次选择`[min_features_to_select, len(feature)]`个特征数量的特征子集进行模型训练和交叉验证，最后选择平均分最高的特征子集。






```python
from sklearn.svm import SVC
svc = SVC(kernel="linear")

from sklearn.model_selection import StratifiedKFold
from sklearn.feature_selection import RFECV
rfecv = RFECV(estimator=svc,           # 学习器
              min_features_to_select=2, # 最小选择的特征数量
              step=1,                  # 移除特征个数
              cv=StratifiedKFold(2),   # 交叉验证次数
              scoring='accuracy',      # 学习器的评价标准
              verbose = 0,
              n_jobs = 1
              ).fit(X, y)
X_RFECV = rfecv.transform(X)
```

随机搜索

简单随即搜索

随机选择多个特征子集，然后分别评估模型表现，选择评估分数高的特征子集

Null Importance

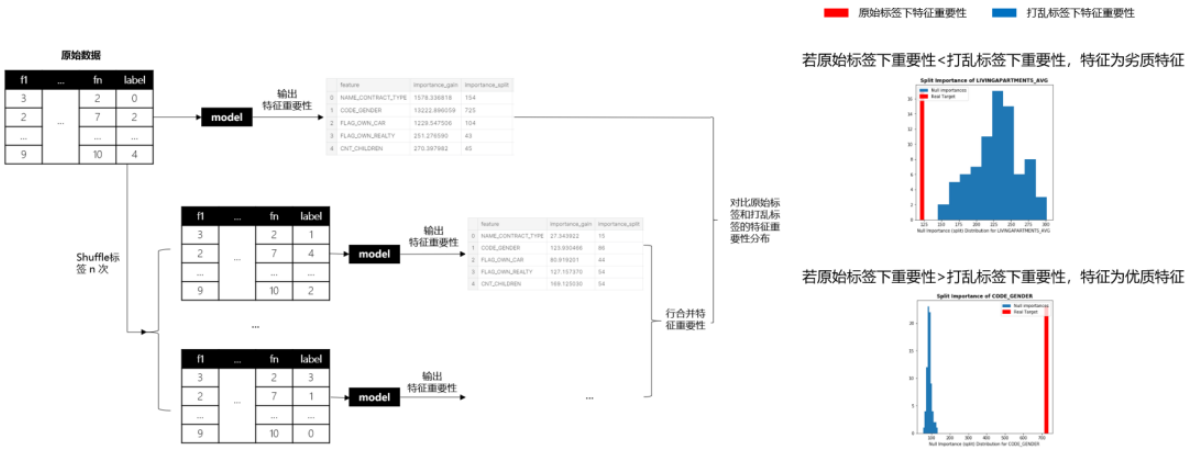
这基于一个简单有效的想法：**真正强健、稳定且重要的特征一定是在真标签下特征很重要，但一旦标签打乱，这些优质特征的重要性就会变差。相反地，如果某特征在原始标签下表现一般，但打乱标签后，居然重要性上升，明显就不靠谱，这类“见风使舵”的特征就得剔除掉。**

例如：

> 如果我们把ID作为特征加入模型，预测不同ID属于哪类消费人群，一个过拟合的模型，可以会学到ID到消费人群的直接映射关系(相当于模型直接记住了这个ID是什么消费人群)，那如果我假装把标签打乱，搞个假标签去重新训练预测，我们会发现模型会把ID又直接映射到打乱的标签上，最后真假标签下，ID“见风使舵”地让都自己变成了最重要的特征。

Null Importance 计算过程大致：

1. 在原始数据集运行模型获取特征重要性（importance_gain/split等）
2. shuffle多次标签，每次shuffle后获取假标签下的特征重要性
3. 计算真假标签下的特征重要性差异，并基于差异，筛选特征



重要性比较一般可采取：

- 分位数比较：

$$\text{Feature Score } = \log\left(10^{-10} + \frac{\text{Importance }_{\text{real label}}}{1 + \text{ percentile }\left(\text{ Importance }_{\text{shuffle label}}, 75\right)}\right)$$

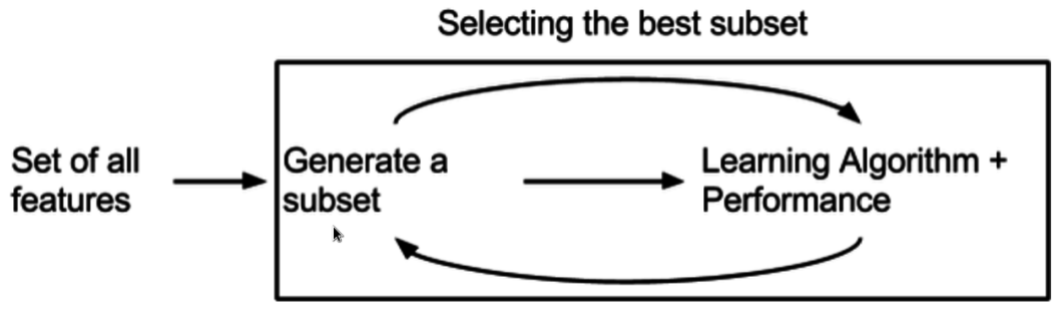
- 次数占比比较：

$$\text{Feature Score} = \frac{\sum((\text{Importance}_{\text{shuffle label}}) < \text{percentile}(\text{Importance}_{\text{real label}}, 25))}{n_{\text{importance}_{\text{shuffle label}}}}$$

总结

在包装法做特征筛选中，可以看出RFECV和Null Importance是适合被考虑的方法

嵌入法



常见的嵌入法有LASSO的L1正则惩罚项、随机森林构建子树时会选择特征子集。嵌入法的应用比较单调，`sklearn`有提供`SelectFromModel`，可以直接调用模型挑选特征。

特征选择总结

综合上述三种特征选择方法考虑，分析各方法优缺点

过滤法

优点

- 非监督，无需提供学习模型
- 效率较高，计算开销小
- 能有效避免过拟合

缺点

- 对后续学习器针对性较弱
- 可能减弱学习器拟合能力

包装法

优点

- 特征选择比过滤法更具针对性
- 基本可以保证有利于模型性能

缺点

- 计算开销更大

嵌入法

优点

- 比包裹法更省时省力，把特征选择交给模型去学习

缺点

- 增加模型训练负担

Summary

对于基学习器为`xgboost(GBDT)`时的因子筛选问题，更多需要关注非线性贡献因子的筛选。

从单因子测试方面看，Rank IC可以提供一定非线性预测能力的参考，相比IC其表示性更强；若融入行业因子与市值因子，回归法也可以考虑被使用，这两因子同样提供了非线性部分；但最具有表示能力的仍是分层回测方法，可以配合交易系统直接挖掘非线性贡献。考虑到分层回测有着最大的计算开销，所以在进行因子分层回测前应先进行高效特征选择。

从基于预测目标的特征选择看，首先应考虑使用过滤法进行低计算开销的初步因子过滤，如频数分析与方差分析，而多变量中可重点关注非线性类，如：Spearman相关系数、Kendall相关系数、卡方检验、互信息分析等。包装法直接作用于目标学习器，利用较高计算开销得到更有效的因子筛选，且具有很强的非线性筛选能力。包装法中考虑RFECV和Null Importance方法，其中后者计算开销相较更小且实际效果优秀（来自Kaggle Grandmaster），适合作前序筛选。与包装法相对，嵌入法直接根据结果进行筛选，然而选取的基学习器较简单如LASSO，所以包装法和嵌入法的计算开销与采用次序需要根据具体问题分析。

总的来说，在启发式筛选之外，基于数据的筛选也十分具有必要性。在最终回测前配合使用多重特征选择方法有助于减轻模型的学习负担，也同样可以增强模型的鲁棒性。