1. factor momentum /multifactor.py
2. 读取数据- 数据预处理：
   1. return：行业/市值中性化
   2. 因子值：可以滤波，降噪比如wavelet方法（可以提高因子的ic）/denosing.py
      1. 注意点：在用滤波器的时候，只能用过去的数据降噪，不能一次性将所有因子值降噪，否则会引入未来数据的影响，比如有的因子和收益率相关时。
      2. 所以要滚动式过滤，每一天都要过滤之前的因子数据
3. 筛选特征，比如xgboost初步筛选，再用factor momentum的回归方法进一步筛选动量稳定的因子
   1. 注意不能用包含未来信息的函数
   2. 每隔一段时间，feature\_collect一次，换一种风格
   3. 筛选因子可能的方法：控制因子风格，比如一半用回归系数绝对值大的（相关性强但收益不一定强），一半用。。。
   4. 当前因子收益率(Y)和过去一年因子收益率(X)之间回归的话，注意当前因子收益率不能用日级数据，可以用月级数据或者周级,但是是固定时间点的计算（比如每月月末）而不是每天的滚动计算，否则很对每天的变动很敏感
4. 估计每天的因子收益率，计算因子收益率矩阵，绘图检查筛选是否有效
   1. 估计方法：每天按因子值排序，最高组的平均收益减去最低组的平均收益
      1. 存在的问题：会出现有的因子每天的因子收益率都是正的，但是实际上策略无法实现多空组合，训练方法与策略方法出现偏差。每天正的收益导致出现极大的累计收益，如果后面因子权重按收益率来分配，会导致其几乎占据了所有权重。
      2. 更新方法：前250天用原因子值，后面的每一天更新。
5. 给每个股票和每个因子分别分配权重
   1. 因子权重
      1. # long\_short: sum (weight<0) = -1/2, sum (weight>0) = 1/2 分方向归一化后，再一起归一化， w有正有负，不作为投资建议，因为无法实现，只能用来判断因子是否有效,正收益说明策略有效
      2. # long: sum abs(weight) = 1, w有正有负 做空的因子变成买底部的，做多的因子变成买顶部的
      3. # long\_only : sum weight = 1, w全正,卖空的因子为0
      4. # short\_only: sum weight = 1, w全正,做多的因子为0
   2. 股票权重：
      1. 等权
      2. 按收益率
6. 计算每天每只股票的权重
7. 回测
   1. 分n个账户，每个账户使用1/n的资金，调仓周期为d天，计算换手率，收益率
   2. 利用quanstats包绘制回测结果和报告
8. 收益分解与策略有效性分析：
   1. 计算不同股票池下的超额收益，进行对比分析比如沪深300或者中证1000
   2. 量化的分层测试：把股票池1分成5个子池子（比如按一段时间收益率分），然后分别看效果
   3. 持仓分析，比如防守股票（基本面好的）比例，进攻股票比例
   4. 用模拟数据（添加了很多涨停的股票，因子里面加了噪声）toy\_data.pkl，如果每天的收益接近0.1，说明把很多涨停的股票挑选出来了，换手率也很低的话，说明策略是有效的。
9. 数据筛选：因子库控制一下，比如60%基本面因子，20%量价因子
10. 减少回撤的原理：当天因子收益率回撤较大的因子，过去一年的累计收益率也会下降，使得当天的因子权重降低，更多权重转移到更稳定的因子去了（逻辑是，选出来的因子都是稳定性比较强的，过去一年的累计收益率下降了，会认为之后会继续下降）
11. 买跌停封手单量大的股票的逻辑：
    1. 如果庄家想要出货，完全可以不封死，甚至可以反复开板出货，或者一开始就不跌停出货。封死是为了制造恐慌情绪，逼迫游资割肉或者加仓止损（绑在一条船上），做空获取收益，后续再拉升出货。