Prez :

page1: self introduction

page2:

We will start by talking about the construction of factors in round 1.

Our main goal in this phase was to develop a model that could predict returns and have a high correlation with the target in the competition.

To do this, we created alpha factors using three different types of factors: In-day period factors, day factors, and N-days factors.

We began with In-day factors, which were built using ticker data. We did some operations such as reports and subtractions for each period to build some factors. For exemple low2high which is calculated by dividing the low by the high.

Next, we built some day factors, we have fundamental market data such as PE, PB, PS, and turnover ratio from the source data.

We built Kline data from the ticker data for each day.

We built some aggregated factors from the In-day factors, such as mean or standard deviation over a 50-periods each day

We built some correlation factors that compute the correlation on 50 ticker data between some factors, for example corr\_ret\_vwap is the correlation between return and vwap from 50 period in a day.

Finally, we moved on to N-day factors. We built the N-days factors from N days. For example, we computed the mean or standard deviation using a 30-day sliding window to demonstrate long-term trends.

Page3:

On this page, we'll discuss our model for the competition. We chose to use lightgbm as our ML model because it's lightweight and widely used in recent financial competitions. We discovered that having more features isn't better for the performance during exploration, and some features were repeated, so we applied feature selection to keep only the most useful ones.

For the first model, we used the SelectKbest class from the scikit-learn package to find the best features. We ended up saving 86 factors for this model. In the second model, we performed additional manual selection and reduce it to just 36 factors.

We also tried a lot of experiments on the hyperparameters of lightgbm. We identified some important parameters, such as setting min\_data\_in\_leaf around 200 and subsample around 0.5 to prevent overfitting.

In the first round of the competition, our model achieved a score of 0.07 on the public leaderboard and 0.083 on the private leaderboard.

Page4:

Let’s move on to round 2, our objective is to build an investment strategy that show a good performance for the next trading days. To achieve this, we plan to use the results of the first round as a signal and additional leverage to improve our score.

When designing our strategy, we considered three key points. Firstly, we found that certain investment codes displayed better correlation with the predicted return, ranging from 0.05 to 0.35, suggesting that our model's signal could perform better on these investments than on others.

Secondly, we observed that frequent switching of investment codes would result in high transaction fees. To reduce these fees, we limited our investment changes.

Thirdly, in our final selection of investment codes, we selected the investments with low correlation among themselves, which means choosing specific investments that differ from the others.

We have illustrated the process with a diagram.

We started with 54 investment codes and computed the mean correlation using a sliding window of 267 to select only the top 10 investments.

Next, we manually selected three investment codes that satisfied the two conditions (high mean correlation to target and low correlation among themselves).

The model will generate signals for these investment codes each day, and we will invest in the code with the highest signal.

Page5:

As for portfolio allocation, we decided to trust our model, so we plan to allocate 99% of our portfolio to the first signal and 1% to the second signal. Alternatively, we could invest 100% of our portfolio in the first signal.

In terms of strategy capacity, if we have a large capital and wish to replicate our strategy, we believe it is possible. We would need to identify other groups of investment codes that cover the two criteria we mentioned earlier: high correlation to the target and low correlation among themselves.

We also implemented some risk management techniques. For instance, we only invested in signals that were between the 30th and 95th percentiles to avoid extreme cases. This approach can reduce drawdowns and improve returns at the same time. So, we did not use this technique in our submission.

Page6:

In the simulation results for my strategy, as shown in the left graph, we were able to achieve a 57% return with a Sharpe ratio of 1.38. However, there was also a 30% drawdown for some period, indicating that the stability of the model needs to be improved.

In the middle graph, we analyzed the position sequence provided by the model and found that we could hold a position continuously for 4-5 days, which reduces our transaction costs. This is because we limited our trading to only three stocks, increasing the probability of selecting the same stock continuously.

In the right graph, we list some indicators of my strategy. Overall, these results demonstrate the potential effectiveness of our strategy, although there is improvement in terms of stability and risk management.

Thank you for listening.

第1页：自我介绍

第2页：

我们先来谈谈第一轮的因素构建。

我们在这一阶段的主要目标是开发一个能够预测收益并与竞争中的目标有高度相关性的模型。

为了做到这一点，我们使用三种不同类型的因子创建了α因子： 日内因素、日间因素和N天因素。

我们从日内因子开始，它是使用股票数据建立的。我们做了一些操作，如每个时期的报告和减法来建立一些因子。例如，low2high是通过低点除以高点来计算的。

接下来，我们建立了一些日因素，我们有基本的市场数据，如PE，PB，PS，和来自源数据的周转率。

我们从每一天的股票数据中建立了K线数据。

我们从日内因子中建立了一些聚合因子，如每天50个周期的平均值或标准差。

我们建立了一些相关因素，计算一些因素之间的50个股票数据的相关性，例如corr\_ret\_vwap是一天中50个时期的回报率和vwap之间的相关性。

最后，我们转向了N天因素。 我们从N天建立了N天的因素。例如，我们用30天的滑动窗口计算平均值或标准差，以显示长期趋势。

第3页：

在这一页，我们将讨论我们的比赛模型。我们选择使用lightgbm作为我们的ML模型，因为它是轻量级的，在最近的金融比赛中被广泛使用。我们发现，在探索过程中，拥有更多的特征并不是更好的表现，而且有些特征是重复的，所以我们应用特征选择，只保留最有用的特征。

对于第一个模型，我们使用scikit-learn包中的SelectKbest类来寻找最佳特征。我们最终为这个模型保留了86个因子。在第二个模型中，我们进行了额外的人工选择，将其减少到只有36个因子。

我们还对lightgbm的超参数进行了大量的实验。我们确定了一些重要的参数，比如将min\_data\_in\_leaf设置为200左右，将subsample设置为0.5左右，以防止过拟合。

在第一轮比赛中，我们的模型在公共排行榜上取得了0.07分，在私人排行榜上取得了0.083分。

第4页：

让我们进入第二轮，我们的目标是建立一个投资策略，在接下来的交易日显示出良好的表现。为了实现这一目标，我们计划将第一轮的结果作为信号和额外的杠杆来提高我们的分数。

在设计我们的策略时，我们考虑了三个关键点。首先，我们发现某些投资代码与预测回报率显示出更好的相关性，从0.05到0.35不等，这表明我们模型的信号在这些投资上的表现可能比其他投资更好。

其次，我们观察到，频繁切换投资代码会导致高额的交易费用。为了减少这些费用，我们限制了我们的投资变化。

第三，在我们最终选择投资代码时，我们选择了相互之间相关性低的投资，这意味着选择了与其他投资不同的特定投资。

我们用一张图说明了这个过程。

我们从54个投资代码开始，用一个267的滑动窗口计算平均相关性，只选择前10个投资。

接下来，我们手动选择了三个满足两个条件的投资代码（与目标的平均相关度高，相互之间的相关度低）。

该模型每天将为这些投资代码产生信号，我们将投资于具有最高信号的代码。

第5页：

至于投资组合的分配，我们决定相信我们的模型，所以我们计划将我们投资组合的99%分配给第一个信号，1%分配给第二个信号。或者，我们也可以将100%的投资组合投资于第一个信号。

就策略能力而言，如果我们有大量的资本并希望复制我们的策略，我们相信这是可能的。我们将需要确定其他的投资代码组，这些代码组涵盖了我们前面提到的两个标准：与目标的高相关性和它们之间的低相关性。

我们还实施了一些风险管理技术。例如，我们只投资于处于第30和第95百分位数之间的信号，以避免极端情况。这种方法可以减少缩水，同时提高收益。所以，我们在提交的文件中没有使用这种技术。

第6页：

在我的策略的模拟结果中，如左图所示，我们能够获得57%的回报，夏普比率为1.38。然而，在某些时期也出现了30%的缩水，表明模型的稳定性需要改进。

在中间的图表中，我们分析了模型提供的仓位序列，发现我们可以连续持有一个仓位4-5天，这降低了我们的交易成本。这是因为我们把交易限制在只有三只股票，增加了连续选择同一股票的概率。

在右图中，我们列出了我的策略的一些指标。总的来说，这些结果证明了我们策略的潜在有效性，尽管在稳定性和风险管理方面有改进。

谢谢