金融大数据期末大作业小组报告

小组成员：

2024100893 郭昭芃

2024100908 金威廷

2024100924 李宇杰

小组代码仓库：

https://github.com/Githubjinwt/Empirical-Asset-Pricing-via-Machine-Learning

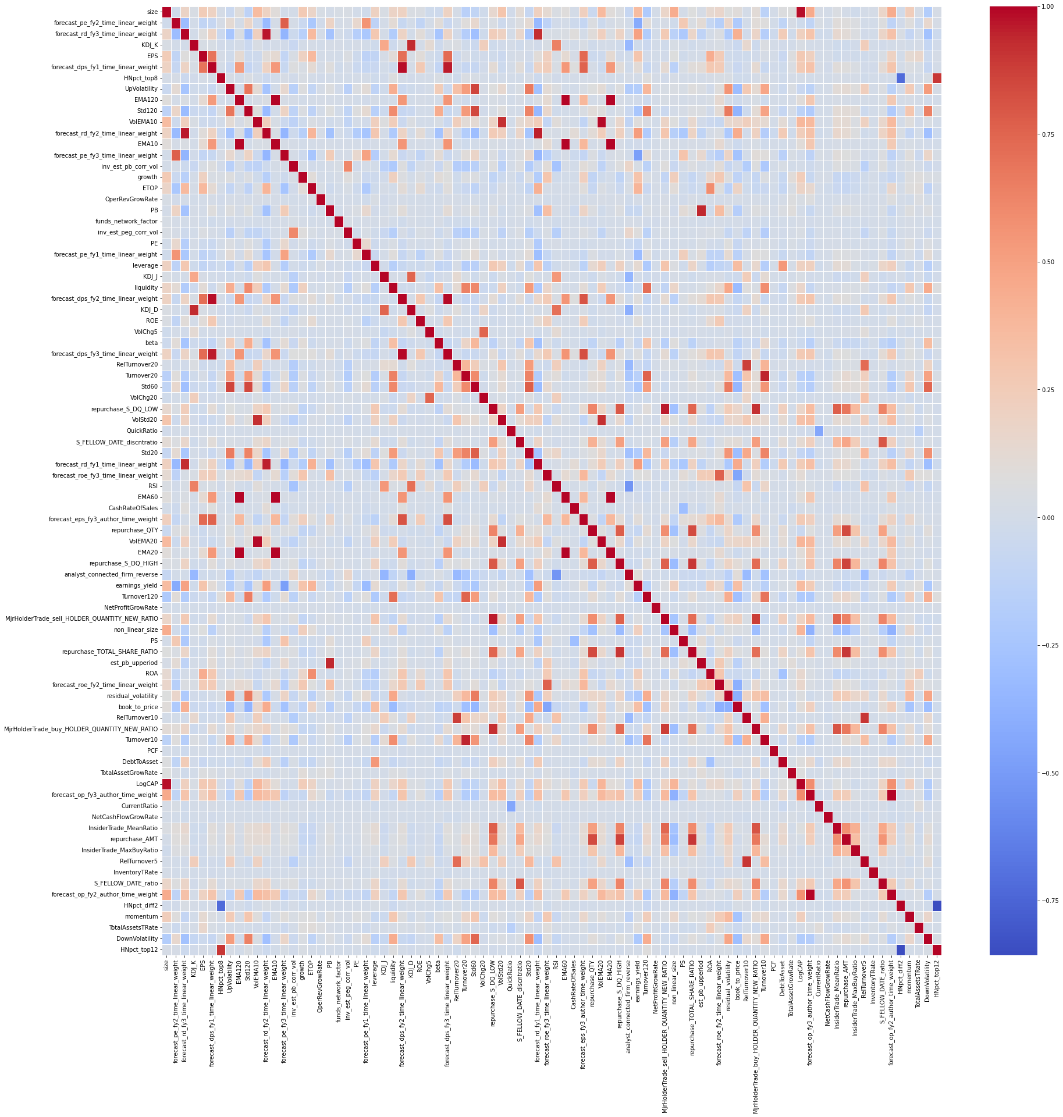
1. 数据

1.1 数据和样本介绍

本小组选取中国A股作为研究样本，数据范围从2015年1月2日至2024年12月31日。共5578支股票，2431个交易日。数据频度为日频（部分报告期披露数据按照数据获取日期ffill到日频）。后续输入模型的数据频度可根据日频数据合成得到。所有数据均来源于WIND和朝阳永续（具体见因子池文件）。

1.2 因子构建

本小组手动构建了共86个因子，包括19个基本面因子、33个另类因子、10个barra因子以及24个低频量价因子（具体见附录）。本小组计算了86个因子之间的相关性，如下图所示。从图中可以看出，因子之间的相关性整体较弱，表明这些因子包含了全面且丰富的信息，能够有效捕捉股票收益的不同驱动因素。低相关性也意味着这些因子在组合使用时能够提供更好的分散效果，从而降低模型过拟合的风险。



1.3 数据预处理

由于不同因子量纲差异较大，本小组对不同因子尝试了CSRank和CSZscore两种标准化方式。对CSRank标准化因子，本小组用中位数填充缺失值，对于CSZscore标准化，本小组用0填充缺失值。

原始因子数据为日频数据。但日频数据可能存在噪音过大的问题，因此本小组同时做了月频数据的模型对比。针对不同类型的因子，本小组采用不同的转换方法：基本面和另类因子：取月末值，能够保留其最新信息。Barra和量价因子：取月均值，能够平滑噪声，反映月度的整体特征。

1. 模型

2.1 普通最小二乘法

普通最小二乘法（OLS）是一种经典的线性回归方法，通过最小化残差平方和来估计模型参数。OLS假设误差项服从正态分布且具有同方差性。然而，当数据中存在异常值时，OLS的表现可能会受到影响。为了增强模型的鲁棒性，本小组使用了Huber回归，它结合了OLS和L1正则化的优点，能够更好地处理异常值。

本小组使用scikit-learn库中的HuberRegressor实现Huber回归。通过网格搜索法优化epsilon（Huber损失函数的阈值参数）和alpha（正则化强度）两个超参数。具体步骤如下：

1. 定义参数网格：epsilon和alpha的取值范围；
2. 遍历所有参数组合，训练模型并计算验证集上的R2分数；
3. 选择最优参数组合，并在测试集上进行预测。

2.2 主成分回归

主成分回归（PCR）通过主成分分析（PCA）将原始特征降维为少数几个主成分，然后在这些主成分上进行线性回归。PCR能够有效处理多重共线性问题，并减少模型的过拟合风险。其核心思想是通过降维保留数据的主要信息，同时去除噪声。

PCR的实现分为两步：主成分分析（PCA）和线性回归。本小组使用scikit-learn库中的PCA和LinearRegression类。通过遍历不同的主成分数量，选择最优的主成分个数。

2.3 偏最小二乘法

偏最小二乘法（PLS）是一种结合了降维和回归的技术。与PCR不同，PLS在降维时同时考虑了自变量和因变量的信息。它通过最大化自变量和因变量之间的协方差来提取主成分，从而在降维的同时保留对因变量的解释能力。

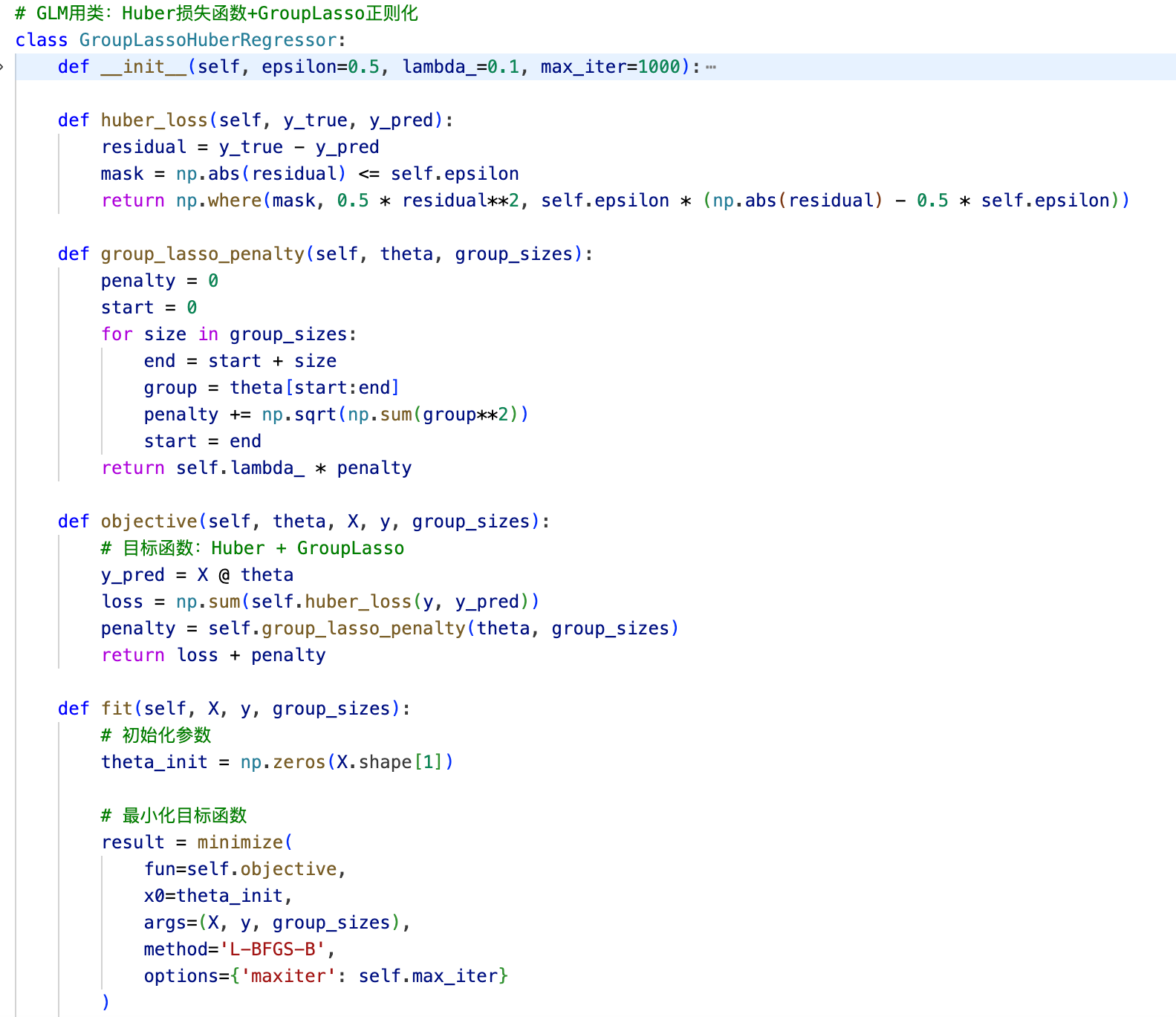
本小组使用scikit-learn库中的PLSRegression实现偏最小二乘法。通过遍历不同的主成分数量，选择最优的主成分个数。

2.4 广义线性模型

广义线性模型（GLM）是线性回归的扩展，允许因变量服从非正态分布。GLM通过链接函数将线性预测器与因变量的期望值联系起来。在本报告中，本小组实现了一个结合样条基函数和Group Lasso正则化的GLM模型。样条基函数用于捕捉非线性关系，而Group Lasso正则化用于增强模型的稀疏性。具体步骤如下：

1. 生成样条基函数：基于分位数确定节点位置，并构建二阶样条基函数；
2. 定义Group Lasso正则化：对每个特征对应的样条基函数进行分组正则化；
3. 使用Huber损失函数和Group Lasso正则化训练模型。

代码：



2.5 弹性网络

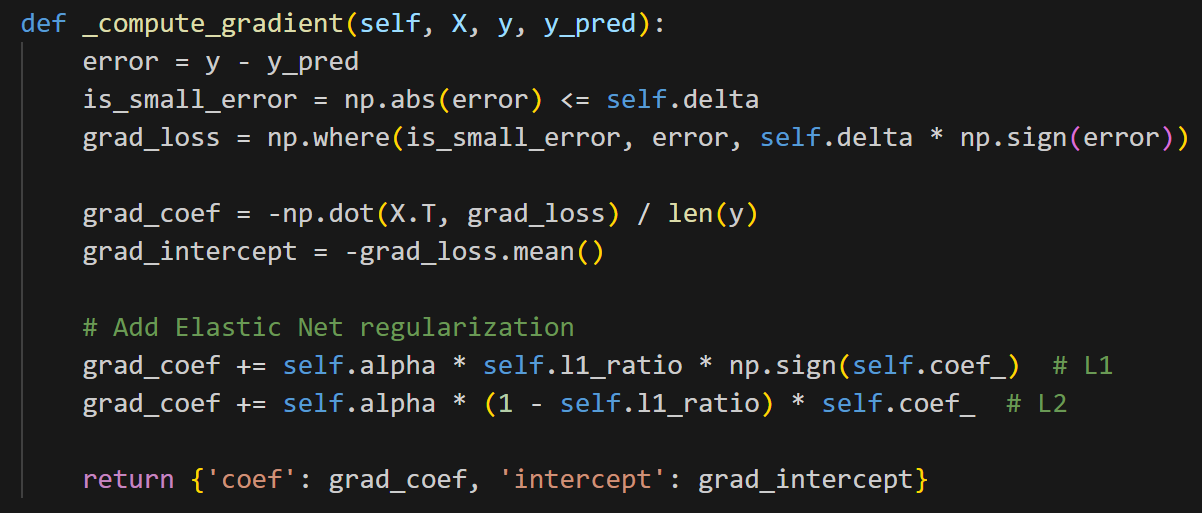
弹性网络（Elastic Net）作为一种线性回归模型，通过结合L1正则化（Lasso）和L2正则化（Ridge），在模型训练中有效平衡了特征选择与模型稳定性。其结合了L1和L2的优点，在特征选择和防止过拟合之间找到更好的平衡点，在高维数据建模中表现优异。

在本小组的模型中，Elastic Net的正则化强度由超参数alpha控制，而L1与L2正则化的比例由l1\_ratio决定。通过这两个超参数的调整，模型能够在控制复杂度的同时，保持对数据中有用信息的有效提取。

本小组结合了Huber Loss与Elastic Net正则化，设计了一个新的回归模型——**HuberElasticNet**。由于scikit-learn中的Elastic Net默认采用MSE损失，并没有直接提供Huber Loss的实现，因此本小组采取了继承和扩展的方式，基于sklearn.base.BaseEstimator和RegressorMixin类重新定义了新的模型类。

为了更好地训练模型，本小组采用了基于梯度下降的优化方法，而非传统Elastic Net的坐标下降法。在每一轮迭代中，通过计算当前预测值与真实值之间的误差，并根据Huber Loss对误差进行分情况计算梯度。损失部分的梯度基于Huber Loss的定义进行更新；正则化部分的梯度则根据L1和L2正则化项的规则进行更新。最终，模型的权重和截距通过梯度下降方法得到更新。模型还设置了最大迭代次数max\_iter和收敛容忍度tol，以便在梯度下降优化过程中控制计算过程。

在\_compute\_gradient方法中，首先计算损失部分的梯度，即误差的变化率，然后计算L1和L2正则化部分的梯度，并将其合并到损失梯度中进行参数更新。具体地，L1正则化项通过符号函数计算梯度，而L2正则化项则直接与系数成正比。代码片段示例如下：



2.6 随机森林

随机森林（Random Forest）是一种基于集成学习的非参数模型，通过构建多个决策树并对它们的预测结果进行平均来降低模型方差，从而提高整体鲁棒性。其核心思想在于利用自助采样（有放回抽样）为每棵树提供不同的训练样本，同时在节点分裂时随机选择部分特征进行决策。这种双重随机化不仅增强了各树之间的差异性，防止单棵树过拟合，还使得模型在处理高维数据和噪声干扰时具有良好的泛化能力。

最初，本小组采用了sklearn.ensemble中的RandomForestRegressor来实现随机森林回归，但在大规模和高维数据场景下，其运行速度和内存占用均表现不尽理想。为解决这一问题，本小组转而采用了LightGBM框架，并将boosting\_type参数设置为rf以模拟随机森林的构建方式。LightGBM通过对连续特征离散化构建直方图，大幅降低了分裂点搜索的计算复杂度，同时其精细的内存管理和多线程并行优化显著提升了训练效率。尽管LightGBM主要以梯度提升树见长，但其灵活的设计同样支持随机森林实现，使本小组能够在保持模型预测精度和稳定性的同时，大幅提高训练速度和资源利用率，更好地满足A股收益率预测任务对实时性和高维数据处理的要求。

2.7 梯度提升回归树

梯度提升回归树（GBRT）是一种通过迭代构建弱预测模型（通常为决策树）的集成方法，其核心思想在于逐步纠正前一轮预测中的误差。在每一轮训练中，模型都会针对当前残差——即真实值与预测值之间的误差——的负梯度方向拟合一棵新的树，从而不断修正前期的不足。与此同时，GBRT支持任意可微分的损失函数优化；在本模型中，本小组选择Huber损失，以兼顾小误差时的平滑性和大误差时的鲁棒性。通过不断累积各树的改进效果，模型逐步逼近目标函数的最优解，体现了集成方法中逐步优化的优势。

最初，本小组计划采用sklearn.ensemble中的GradientBoostingRegressor实现GBRT+Huber模型，但在大规模、高维数据的实际训练中，发现其存在明显的计算瓶颈，导致训练速度过慢。为此，本小组转而采用LightGBM框架，并将关键参数设置为boosting\_type='gbrt'和objective='huber'，明确利用梯度提升树构建方式并直接使用Huber损失。相比于传统实现，LGBM通过高效的直方图算法、优化的内存管理以及多线程并行计算，大幅提升了训练效率。其具体性能优势已在前述模型中详细讨论，此处不再赘述。

2.8 神经网络

全连接神经网络（Neural Network）由多个层次的神经元构成，每一层神经元均与下一层所有神经元全连接，从而实现输入信息的加权组合和非线性变换。具体而言，每个神经元对输入进行加权求和并加上偏置，再通过激活函数（例如ReLU）进行非线性映射，使得网络能够捕捉复杂的输入与输出之间的关系。通过反向传播算法，将损失（如均方误差）关于各层参数的梯度逐层反馈，并利用自适应优化器（如Adam）更新参数，从而使得网络逐步逼近最优解。

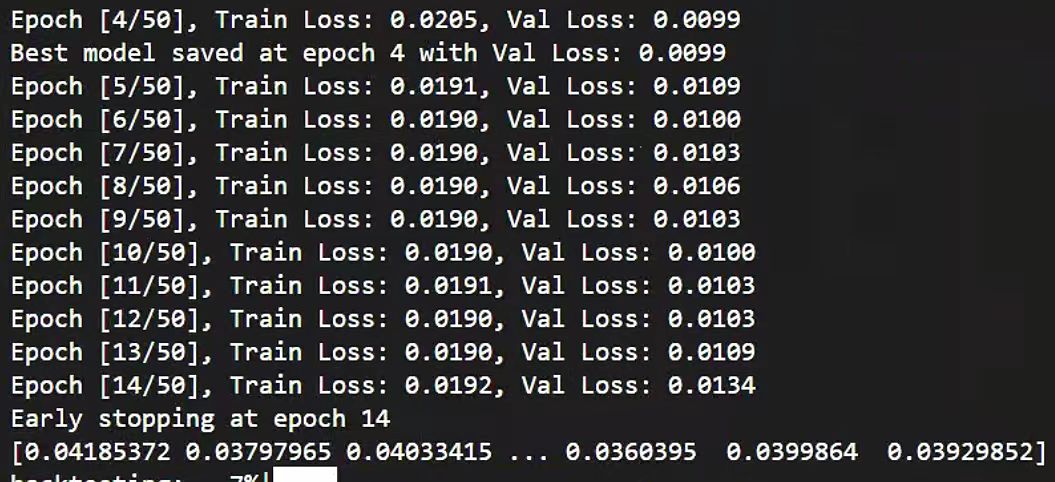
1. 设计方案

在Gu等的工作中，原文对比了2至5层全连接神经网络在预测任务中的表现。受限于GPU资源以及A股数据噪音较大的现实情况，本小组选择构建了一种简化的两层全连接结构。尽管较深层网络理论上能够捕捉更丰富的特征表达，但实际应用中深层模型不仅计算开销更大，而且在高噪音环境下容易导致过拟合问题。

因此，本小组在保证计算效率和稳定性的前提下，专注于两层网络的研究。该网络采用ReLU激活函数，以提高非线性表达能力，并利用Adam优化器实现自适应梯度更新，期望在简化结构下仍能提取到足够的特征信息，从而实现合理的预测性能。本小组的设计充分考虑了资源约束与数据特性之间的平衡，旨在通过精细调参和网络结构的合理设计，为后续深入改进和性能提升奠定基础。

1. 遇到的问题

在初始训练过程中，模型在验证集上的loss始终未能下降，导致对未见数据的泛化能力受到严重限制，预测效果不尽如人意。进一步观察发现，训练过程中的损失函数变化极为缓慢，且模型在不同迭代阶段几乎没有取得实质性改进，这为模型的稳定性和收敛速度带来了挑战。（本部分的的模型结果保存为NN2）



1. 原因分析

经过本小组讨论分析，这种现象可能源自多个方面：

首先，模型参数初始化后可能落入鞍点或梯度平坦区，使得梯度更新非常缓慢，难以突破局部最优；其次，A股数据本身噪音较大，噪音信号可能干扰了有效特征的捕捉，导致梯度方向不稳定；此外，采用简化的两层结构虽然降低了计算资源要求，但在处理复杂、高噪音数据时，其表达能力可能不足，无法提取出稳定且具有区分性的特征信息。

综上，尽管本小组基于资源和数据的限制选择了较为简单的两层结构，并采用了成熟的ReLU和Adam优化器，但初始训练中验证集loss不降的问题表明网络在参数空间中存在优化瓶颈和数据干扰。后续工作将针对这些问题，探讨更高效的初始化策略、数据预处理方法以及网络结构调整，以提升模型的收敛速度和泛化能力。

1. 模型提升

3.1 全连接神经网络的改进

在初始版本中，全连接网络在训练过程出现验证集loss长时间无法下降的现象。针对这些现象，本小组采取了如下改进策略：

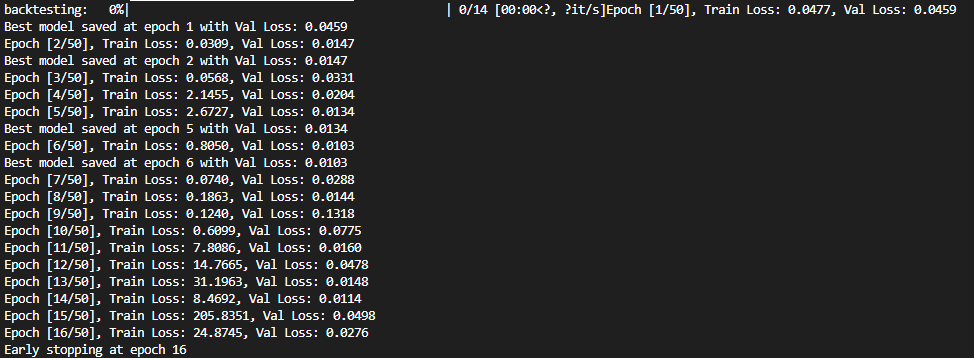
1. 激活函数和学习率调整

为缓解ReLU可能出现的梯度消失问题，本小组将激活函数由ReLU替换为LeakyReLU。LeakyReLU在负输入区域保持一个较小的非零斜率，有助于梯度流动，从而使网络更容易摆脱梯度平缓区域。虽然调整的代价是让计算梯度耗时更长，但整体来看却是较有效地缓解了梯度消失问题。同时，本小组适当调大优化器学习率，使得参数能够更快地走出梯度较小的鞍点或平坦区域，进一步促进损失下降。

1. 数据预处理策略优化

在股票因子数据处理中，本小组最初对缺失值采用了 fillna(0) 的简单填充方式。考虑到零值填充可能会扭曲数据分布和特征统计，本小组尝试使用均值填充，以期提供更合理的缺失数据估计，从而降低噪音对梯度计算的干扰。

尽管上述改动缓解了训练集loss停滞的问题，但随后观察到，在少数几次迭代中，train loss波动非常剧烈，甚至超过了1，甚至达到了几十、几百。但预测的收益率是小于1的小数，损失函数是MSE，正常来说根本不应该出现这样的loss。



经分析，本小组将之归咎于某些数据局部梯度陡峭甚至梯度爆炸。为此，在模型初始化阶段采用了更为精细的权重初始化策略，并在反向传播时对梯度进行裁剪，将梯度值限制在[-5, 5]范围内。这一举措有效防止了梯度爆炸，确保了参数更新的稳定性，使得网络训练过程趋于平稳。这一结果最终在回测中记录为NN2\_2。

**进一步说，**上述改动主要是为了使模型能够“正常”运行，并非真正提升模型泛化能力。为进一步抑制过拟合风险，本小组引入了Dropout机制及额外的正则化项，同时将Adam优化器替换为AdamW。AdamW对于Adam的改进主要源自于Adam的L2正则化问题。大多数优化器通过引入weight decay来实现L2正则化。但事实上Adam+weight decay并不能实现L2正则化效果，甚至会出现不理想的迭代结果：

若Adam加上weight decay，那么梯度的结果为：。表示第t次迭代中更新的神经网络参数；为损失函数，表示在参数位置的损失函数；第t次迭代中损失函数的梯度，即为一阶导函数；表示学习率，表示权重衰减(weight decay)系数/正则项系数。在此基础上根据求解一阶动量和二阶动量：，为Adam优化器的平滑参数(用于动量指数平滑)。将上式代入参数更新公式：



为防止优化器算法除0而设置的保护参数。此处L2正则化的本意在于使得参数有着更小的值从而解决过拟合，但上式中，L2正则化系数的加入使得分子变大，但分母的二阶动量同样也受到正则化系数的影响，从而使得系数的更新变化变得模糊，正则化效果可能会被削弱甚至出现负面效果。

对此，AdamW做出技术改变，将weight decay惩罚项从损失函数梯度更新中分离，并直接把其加入到参数更新中：



这样可以直接做到L2正则化的效果，并且不会造成结果的负面影响，从而一定程度上缓解了模型训练的过拟合问题。这一结果在最终回测中记录为NN2\_3。

3.2 基于Transformer的股票因子数据建模

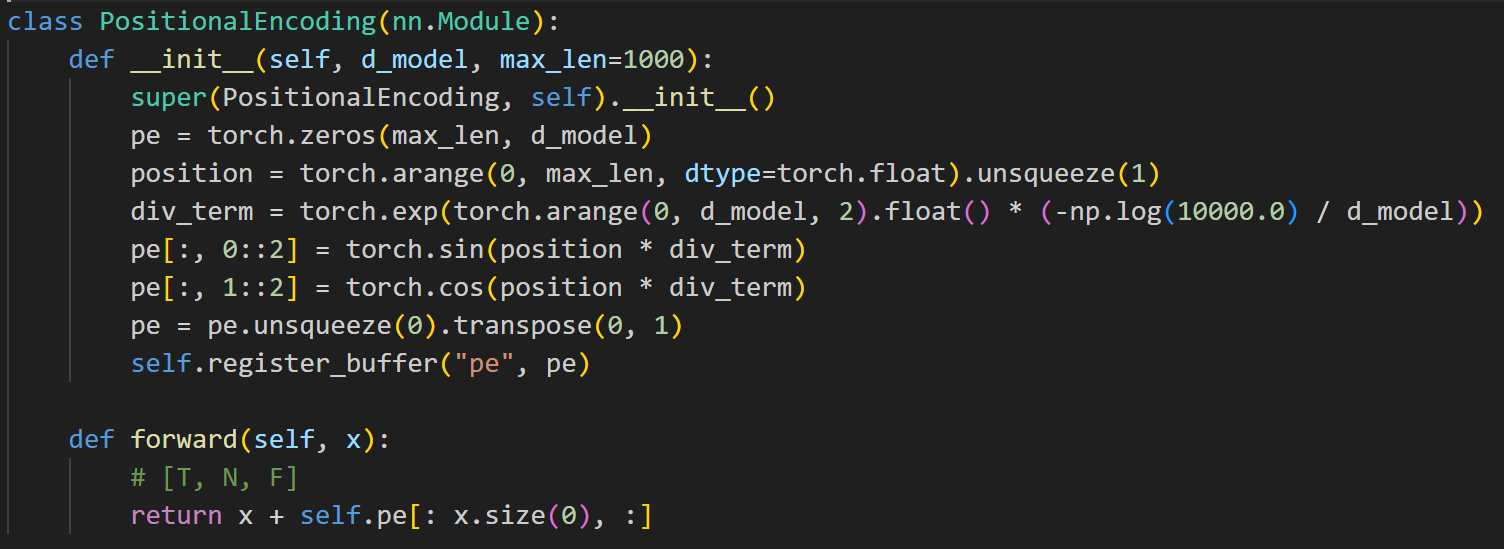
为进一步提升股票因子数据的建模能力，本小组引入了Transformer架构。Transformer由Vaswani等人提出，其核心在于自注意力（Self-Attention）机制，能够捕捉输入数据中各部分之间的全局依赖关系。相比传统的全连接网络或循环神经网络，Transformer不仅在建模长距离特征交互方面更为精确，而且天生支持高效并行计算，从而大幅加速模型训练。

在本小组的探索中，Transformer模型的基本设置与改进后的全连接网络保持一致，主要包括采用AdamW优化器以确保参数更新的稳定性并强化正则化效果，以及保持相同的激活函数和基础超参数，从而在对比实验中消除其他变量的干扰。通过这种设计，本小组期望利用Attention机制更深层次地挖掘股票因子之间的复杂交互信息，进而提升预测的准确性与鲁棒性。

为实现该目标，本小组继承torch.nn.module模块，按照Transformer最初的设计构想，自定义实现了以下关键类函数：

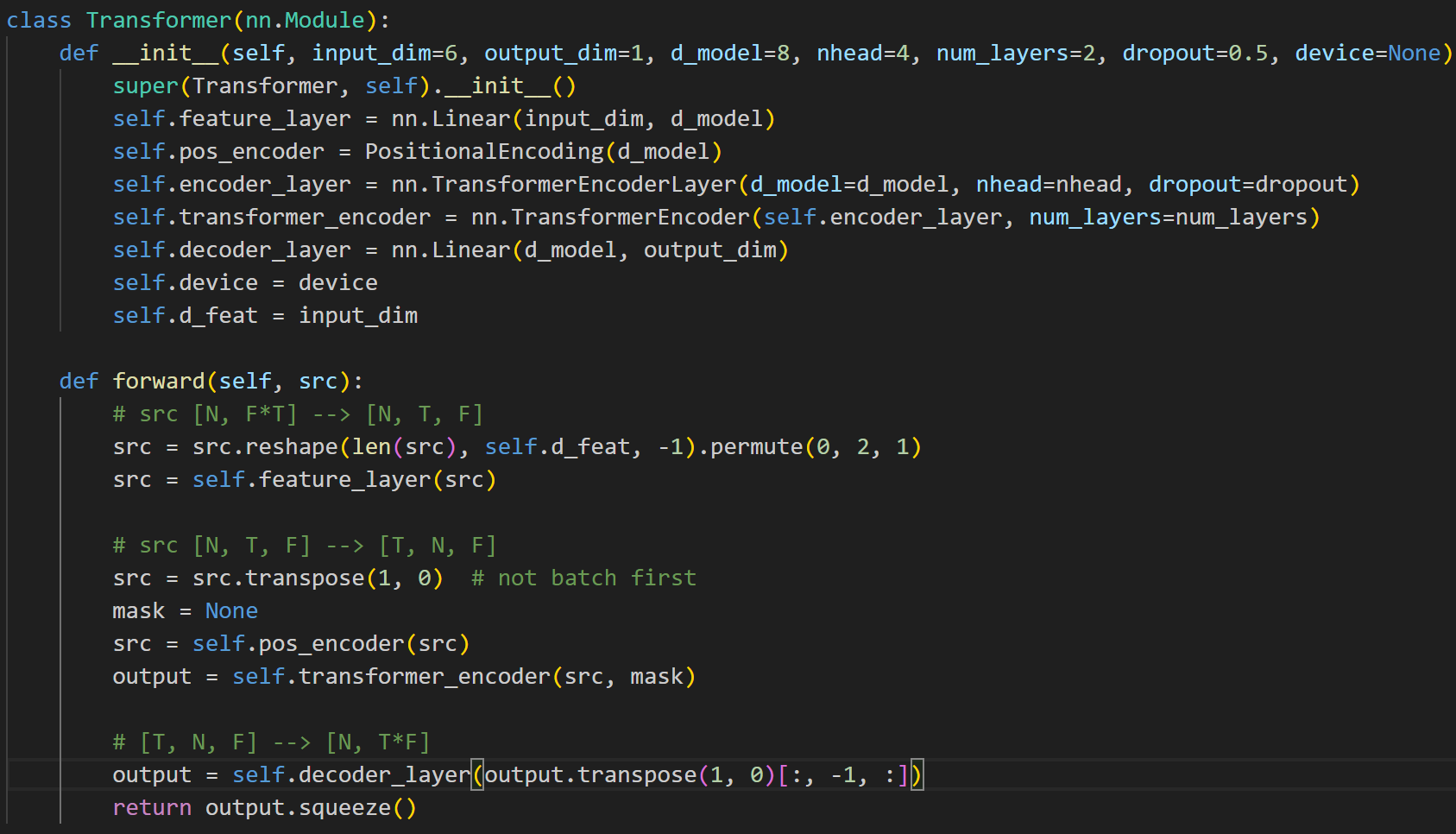
1. Positional Encoding模块

Transformer的并行处理特性使其缺乏序列顺序信息，因此需要引入位置编码。该模块通过对位置向量应用正弦和余弦变换，将位置信息编码到输入特征中。具体而言，构造函数中生成一个形状为(max\_len, d\_model)的编码矩阵，并分别对偶数和奇数维度应用正弦和余弦函数。其forward方法则将输入张量与相应位置编码相加，使模型能够区分不同位置的特征。



1. Transformer模型实现

模型首先通过一个线性层将输入数据映射到d\_model维度，然后经过reshape和transpose调整为Transformer所需的序列格式。接下来，经过PositionalEncoding模块加入位置信息后，数据传入由多个TransformerEncoderLayer组成的编码器，利用自注意力机制提取全局特征。最后，通过一个线性解码层将编码后的特征映射至预测输出，并对输出进行适当变换以获得最终结果。



总体而言，本小组通过引入Transformer架构和相关位置编码机制，力图在股票因子数据中捕捉更丰富的全局依赖和特征交互信息。该模型设计不仅在结构上保持了与全连接网络实验的一致性，也充分利用了Transformer在并行计算和长距离依赖建模上的优势，为提升预测精度与鲁棒性提供了新的思路和方法。

3.3 GBRT的改进

1. 在LGBM中的正则化与随机采样改进

在最初的LGBM实现中，本小组采用了GBRT框架并使用Huber目标函数，但发现模型存在过拟合风险，并容易受到噪音影响。为了解决这一问题，本小组在调参过程中引入了subsample和feature\_fraction参数。这两个参数的设置旨在增加模型的偏置，从而减少模型在训练过程中对噪音的过度拟合。通过只使用部分数据（subsample）以及部分特征（feature\_fraction）来训练每一棵树，模型能够在学习残差时避免捕捉到过多的噪音信息，从而提升预测的稳定性。

与此同时，为了在防止过拟合的同时使模型学习得更精细，本小组调整了其他关键超参数。本小组选择调大n\_estimators，并相应地调小学习率。这一组合策略使得模型在迭代过程中能够以更小的步长逐步优化预测结果，对残差的拟合更加细致，从而进一步提高整体学习效果和预测精度。

1. 从LGBM到XGBoost的迁移

在第一阶段改进的基础上，为进一步提升训练速度和预测精度，本小组将模型从LGBM切换至XGBoost。XGBoost不仅延续了梯度提升框架的优势，还具备许多针对大规模高维数据优化的特性。XGBoost原生支持GPU加速，这在大规模数据下可以显著缩短训练时间，对高维特征和大量训练样本的股票收益率预测任务尤为关键。虽然Lgbm也有GPU加速的接口，但配置环境比XGBoost麻烦很多，成本比较高，在适配度上相对弱于XGBoost。

此外，XGBoost在特征切分上采用了更精细的算法。相比于LGBM，XGBoost能够更准确地找到最佳切分点，充分挖掘数据中的非线性关系和细微特征变化。这种精细的特征切分机制使得每棵决策树在拟合残差时能够更有效地利用信息，进一步提高整体模型的预测效果。

1. 验证样本分割与调参

4.1 样本分割方法

为了确保模型的预测性能能够在样本外数据上得到有效验证，本小组参考文献中的时间序列分割方法，将样本划分为三个不相交的时间段：训练集、验证集和测试集。这种划分方法不仅保持了数据的时间顺序，还模拟了模型在实际应用中的样本外测试场景。其中，**训练集**用于根据一组特定的调优参数值估计模型；**验证集**用于调优超参数，基于训练集估计的模型对验证集进行预测，并根据预测误差计算目标函数，迭代搜索最优超参数；**测试集**既不用于模型估计，也不用于超参数调优，是真正的样本外数据，用于评估模型的最终预测性能。

通过这种划分方法，本小组能够有效避免模型在训练和调参过程中对测试数据的过拟合，从而更准确地评估模型的泛化能力。

4.2 时间序列滚动划分

为了充分利用时间序列数据，本小组采用滚动窗口的方法对数据集进行划分。具体划分逻辑为：

1. 从数据集的起点开始，依次划分训练集、验证集和测试集；
2. 每次划分后，将窗口向前滚动roll\_freq个月，重复上述划分过程，直到测试集的结束日期超出数据范围。
3. 回测

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 年化收益率 | 夏普比率 | 最大回撤 | 交易次数 | IC |
| preds\_XGB | 0.1941 | 0.7842 | 0.4330 | 186955 | 0.0991 |
| preds\_LGBM | 0.1882 | 0.7783 | 0.4241 | 195860 | 0.0967 |
| monthly\_preds\_GLM | 0.1877 | 0.7405 | 0.2910 | 21377 | 0.0488 |
| monthly\_preds\_XGB | 0.1831 | 0.7510 | 0.2380 | 21957 | 0.0726 |
| monthly\_preds\_Transformer | 0.1793 | 0.7307 | 0.2574 | 19679 | 0.0591 |
| monthly\_preds\_ENET | 0.1778 | 0.7296 | 0.2520 | 16018 | 0.0359 |
| monthly\_preds\_PLS | 0.1777 | 0.7074 | 0.2657 | 15604 | 0.0423 |
| monthly\_preds\_RF | 0.1747 | 0.9194 | 0.2630 | 13178 | 0.0399 |
| preds\_LGBM\_RF | 0.1652 | 0.7195 | 0.3271 | 99593 | 0.0747 |
| preds\_PLS | 0.1642 | 0.7494 | 0.3552 | 132853 | 0.0249 |
| preds\_GLM | 0.1635 | 0.6864 | 0.3999 | 204830 | 0.0431 |
| monthly\_preds\_OLS | 0.1614 | 0.6775 | 0.2249 | 21667 | 0.0434 |
| monthly\_preds\_LGBM | 0.1614 | 0.6546 | 0.2889 | 18564 | 0.0221 |
| preds\_OLS | 0.1577 | 0.6359 | 0.3780 | 152449 | 0.0412 |
| preds\_Transformer | 0.1439 | 0.6693 | 0.3152 | 131800 | -0.0082 |
| preds\_ENET | 0.0937 | 0.5397 | 0.3869 | 92783 | -0.0397 |
| monthly\_preds\_NN | 0.0593 | 0.3507 | 0.4124 | 21587 | 0.0301 |
| preds\_NN2\_3 | -0.0133 | -0.0320 | 0.4793 | 212837 | 0.0236 |
| preds\_NN2\_2 | -0.0815 | -0.5046 | 0.5692 | 231425 | 0.0203 |
| preds\_NN2 | -0.1119 | -0.8303 | 0.6494 | 243238 | -0.0043 |

在回测结果中可以观察到，XGBoost在年化收益率上优于LightGBM，尤其是在月频数据上表现更为明显，而在日频数据上二者的差距较小。这一现象可能反映了模型策略与数据特征之间的互动机制。XGBoost采用的是Level-wise分裂策略，相对而言在面对特征较复杂或噪音较大的数据时，其灵活性可能不足，导致拟合效果受限。而日频数据通常数据量大、因子较为有限，这种情形下模型对噪音的敏感性被进一步放大，从而使得XGBoost和LightGBM之间的差异不如月频数据上明显。

另外，回测结果显示，所有基于梯度下降方法的模型表现均不理想，这与树模型、以及OLS等线性模型使用的损失函数计算和参数更新方式形成了鲜明对比。传统方法通常采用闭式解或坐标下降等方式，更新过程较为稳定，而梯度下降法则依赖于全局损失函数的梯度信息。在噪音较多且特征复杂的数据环境中，梯度下降法容易陷入鞍点、平坦区甚至遭遇梯度爆炸问题，导致参数更新不稳定，最终使得模型难以捕捉到数据中的有效信号。

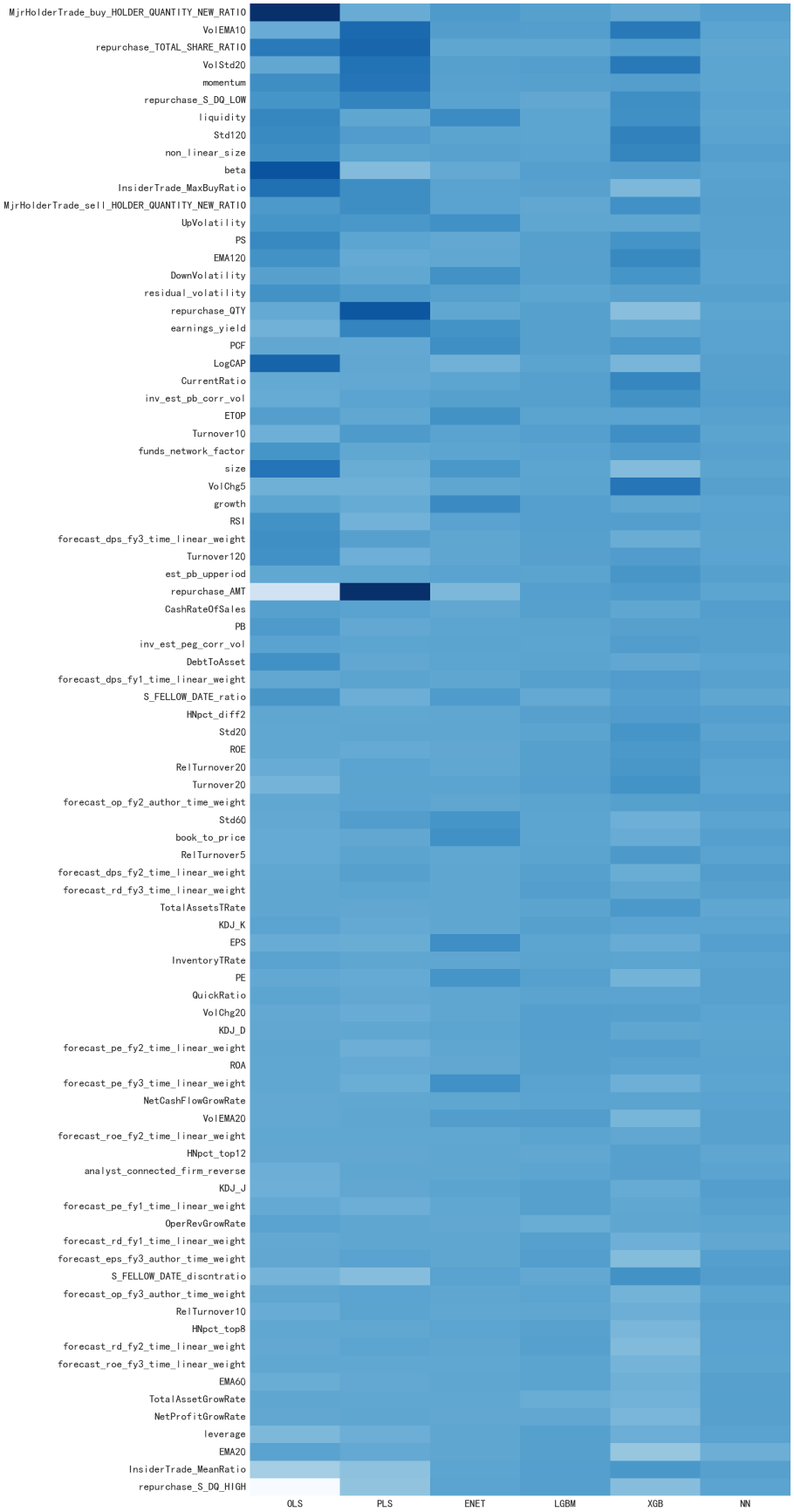
具体来说，A股日频数据通常具有较高的复杂度和噪音水平，通常包含大量微观波动和异常值，这使得基于梯度下降的模型在训练时更容易出现局部最优或梯度不稳定的情况，正如在上文提及的全连接网络的训练过程中，不同时段的股票数据就曾分别出现了梯度消失或梯度爆炸等问题，进而导致了模型的优化瓶颈。而月频数据由于数据聚合效应，相对平滑，噪音减少，因此梯度下降法在月频数据上表现较为稳定，拟合效果也相对更好。

值得一提的是，这并非典型过拟合或典型欠拟合，前者反映的是训练阶段表现出色但测试集拉垮，后者是训练和测试阶段都表现不佳，而本组神经网络训练时更多是遇到Loss卡住无法下降，或是梯度剧烈波动，遇到了较显著的优化问题，或反映出较为复杂的数据特征现象。结合以上现象，可知在数据特征较复杂、噪音较多的情况下，梯度下降法更容易受到局部极值和梯度爆炸的影响，进而影响模型的整体表现；而数据较平滑时，这些问题可以得到一定缓解。

从更深层次看，这也反映出一个因子问题：即现有因子可能无法充分解释日频数据中的所有变动，使得模型在追求精细拟合时更易拟合噪音。而月频数据由于平滑处理减少了噪音，因子解释力相对更高，使得梯度下降法的劣势不那么明显。总体而言，其他模型（如树模型、随机森林、OLS）表现正常，进一步印证了数据特征和噪音水平对基于梯度下降方法的模型训练具有较大影响。在这种情形下，传统模型由于其较为稳健的参数更新策略，能更好地抵御数据中的噪音和复杂性，而梯度下降法则需要更加精细的调参和更高质量的因子才能发挥其优势。

1. 因子重要性

本小组还研究了单个因子对于每个模型性能的相对重要性（ENET、NN模型由于拟合效果欠佳，导致因子对模型的贡献均不显著），最终的结果如下图。本小组对单个因子在不同模型中的平均重要性进行排序，顶部为对所有模型均相对重要的因子，底部为对所有模型均相对不重要的因子，因子重要性越高，颜色越深。



根据结果可以看出，不同的因子在不同模型中的相对重要性存在差异，比如对于beta因子，beta在OLS模型中的重要性显著高于其他模型。这可能是因为OLS是一个纯线性模型，能够直接捕捉beta与目标变量之间的线性关系。而在LGBM、XGB等非线性模型中，beta的重要性较低，可能是因为这些模型更擅长捕捉复杂的非线性关系，beta的线性贡献被其他非线性效应所稀释。这种不同模型间的不一致性也与Gu等的结论相冲突，可能是由于数据样本或因子定义的差异所致。此外，目标变量的特性或模型复杂度也可能影响了因子重要性的评估。未来的研究可以通过统一数据样本和因子定义，进一步验证这些差异的原因。

1. 总结与思考

本小组以A股市场为研究对象，构建了多因子量化策略，并通过多种机器学习模型进行收益预测与回测验证。因子构建中，本小组除传统的量价、基本面因子外，还创新性地引入多种另类因子，捕捉市场动态与事件影响；模型方面，采用XGBoost和LightGBM提升训练效率，并通过正则化与随机采样抑制过拟合。神经网络优化中，引入LeakyReLU和AdamW缓解梯度问题，Transformer则通过自注意力机制捕捉因子间的全局依赖关系。回测结果显示，XGBoost在月频数据上表现最优，而神经网络在日频数据中受噪声影响较大。因子重要性分析表明，不同模型对因子的依赖程度存在显著差异，线性模型更依赖传统因子，而非线性模型通过复杂交互稀释单一因子贡献。本次实践深化了对量化投资系统化方法的理解，未来可进一步探索前沿技术与更高频数据的结合，提升策略的预测能力与稳定性。