基于 XGBOOST 方法预测快照数据股价移动方向

郭小通 2024213253

建模思路

XGBoost（Extreme Gradient Boosting）是一种高效的梯度提升算法，广泛应用于分类、回归和排序等机器学习任务。它基于梯度提升树（GBDT）方法，但在实现上进行了优化，提升了计算速度和模型性能。XGBoost 通过构建多个决策树，并利用梯度提升（Gradient Boosting）算法逐步优化模型，最终形成一个强大的预测器。

• XGBoost 的核心思想

1）集成学习：XGBoost 通过构建一系列决策树，每棵树在前一棵树的基础上进行修正，逐步减少预测误差。每一棵树的构建目标是最小化前一轮模型的残差。

2）正则化：XGBoost 在优化过程中加入了正则项，以控制树的复杂度，防止模型出现过拟合，提高泛化能力。正则化项使得模型更稳定，并在某些情况下能够提升预测精度。

3）贪心算法：XGBoost 使用贪心算法进行树的分裂选择，优先选择能够最大化信息增益的特征进行分裂，从而提高每棵树的预测能力。

• XGBoost 的优势

1）高效性：XGBoost 采用了高度优化的算法，可以充分利用计算资源，在大规模数据集上表现出色。其内置的并行计算能力和硬件加速支持使得它在处理复杂问题时非常高效。

2）可解释性：与深度学习模型不同，XGBoost 提供了良好的可解释性。它不仅能够输出特征的重要性排序，还可以为每一棵树的节点分裂提供具体的决策依据，使得我们可以直观地理解模型的决策过程。

3）处理缺失值的能力：XGBoost 可以自动处理数据中的缺失值。在训练过程中，它会自动学习最佳的分裂方向来处理缺失数据，减少了对数据清洗和预处理的依赖。

4）强大的模型调优能力：XGBoost 提供了多种超参数，可以灵活地调整模型的表现，如学习率、树的深度、正则化参数等。此外，它还支持早期停止（early stopping），避免了过度训练。

• XGBoost 的工作流程

1）初始化模型：开始时，模型的预测值为常数（通常为训练数据的平均值）。

迭代优化：通过计算每个样本的残差，逐步构建新的树来拟合这些残差。每棵树的预测值都会加到现有模型的预测结果上。

2）模型合并：每次迭代都会更新模型，合并新的树模型来提升整体预测能力。

3）最终预测：经过多次迭代，最终得到多个树模型，模型的预测结果是各树结果的加权和。

• XGBoost 的应用场景

1）分类问题：如二分类、多分类问题（例如，癌症检测、图像分类等）。

2）回归问题：如房价预测、金融风险评估等。

3）排序问题：如搜索引擎的排名优化、推荐系统等。

总之，XGBoost 以其高效性、灵活性和强大的性能在机器学习领域获得了广泛应用，特别是在 Kaggle 等数据科学竞赛中取得了显著成绩。

任务理解

• 数据来源

数据包括 10 只股票 79 个交易日的快照数据。

• 模型任务

本次模型任务是预测股票中间价的移动方向，lable5/10= $\begin{cases} 0 & x <- 0.05\% \\ 1 & -0.05\% =< x =< 0.05\% \\ 2 & x > 0.05\% \end{cases}$，

lable20/40/60=$\begin{cases} 0 & x <- 0.1\% \\ 1 & -0.1\% =< x =< 0.1\% \\ 2 & x > 0.1\% \end{cases}$

• 评分标准

1）F0.5

$$\text{precision} = \frac{true\ positive}{true\ positive + false\ positive} = \frac{true\ positive}{no.of\ predicted\ positive}$$

$$\text{recall} = \frac{true\ positive}{true\ positive + false\ negative} = \frac{true\ positive}{no.of\ actual\ positive}$$

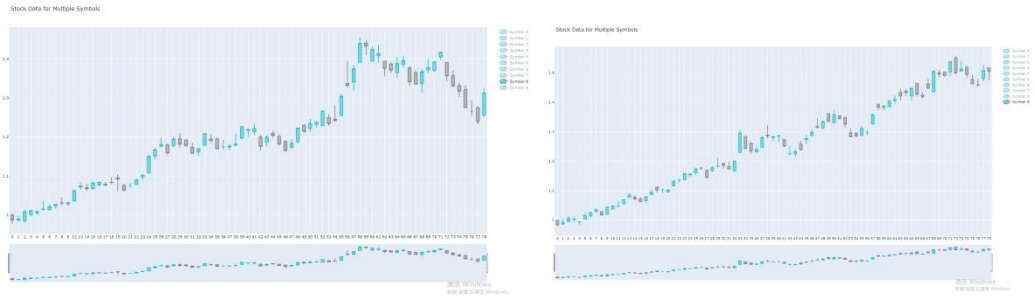
$$F_\beta = (1+\beta^2) \times \frac{precision \times recall}{\beta^2 \times precision + recall}$$

2）P&L

- 对于N-tick的预测模型（N=5,10,20,40,60），
  - 每当模型预测为下跌时：
    - 做空（卖出）1个单位的股票标的，持有N个tick后平仓（买进），记录本次交易收益率
  - 每当模型预测为上涨时：
    - 做多（买进）1个单位的股票标的，持有N个tick后平仓（卖出），记录本次交易收益率
  - 累计所有交易结果，计算整体收益率

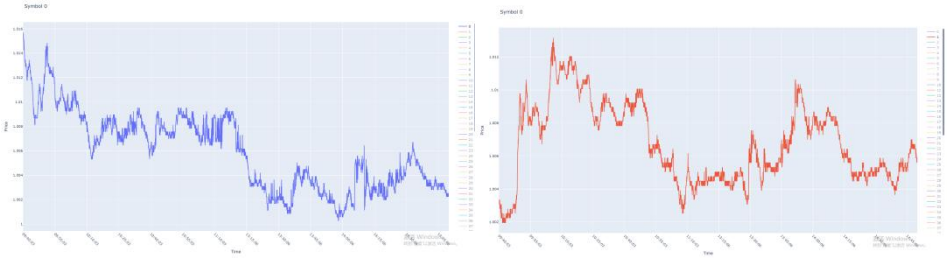
可视化处理

- 日 K 线还原



• **分时图还原**

还原了分时图，观察特征，考虑是否有必要剔除一部分波动率不大的交易时段，观察之后认为无需剔除固定时段。分时图举例如下所示。



数据处理

剔除涨跌停期间数据，剔除缺失值。在全部数据中，21 天缺少数据，一共 90*79 天数据，790 天-21 天 * 99 = 76131 数据删除，最终剩余 2929517 条数据。

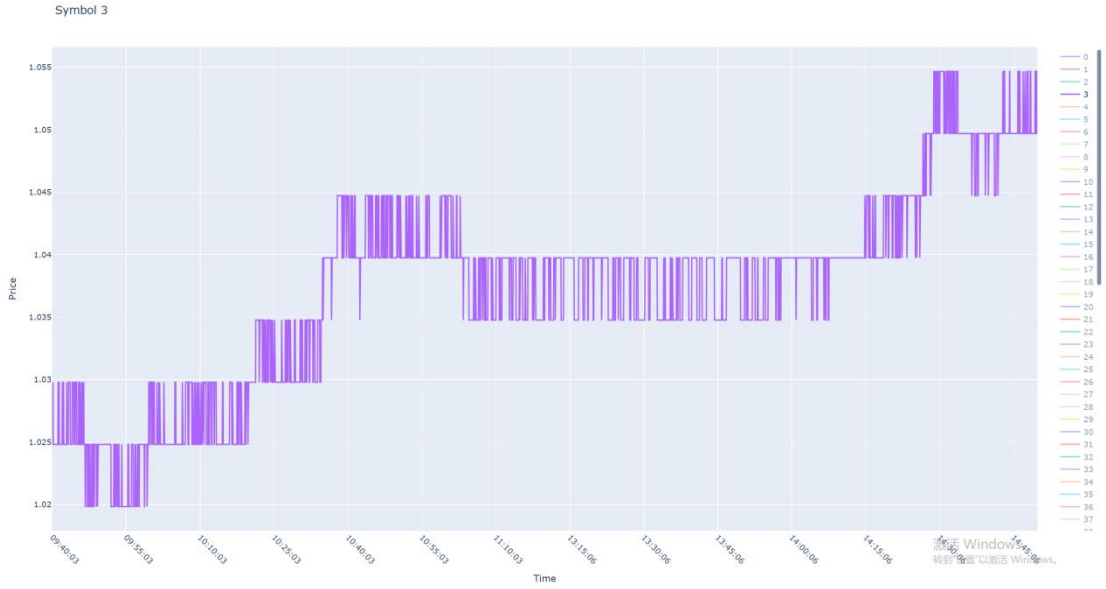
• **分区处理**

用“0.01/最小波动率”的方法估计每只股票的价格，区分出低价股（低于 5 元）和中等价格股票如下图，结果与日分时图线所示相符（低价股分时图“颗粒度”明显），做了区分训练后发现模型效果降低了，所以后来选择用没有区分估计的模型。

A	B	C	D	E
date1	sym	min vol	0.01/（min vol-1）	
	0	1.00027928	35.80635921	
	1	1.003861	2.59000259	
	2	1.00034412	29.05963036	
	3	1.005025	1.990049751	
	4	1.00084317	11.86000451	
	5	1.0011236	8.8999644	
	6	1.00119403	8.374998953	
	7	1.00089767	11.13995121	
	8	1.0007485	13.36005344	
	9	1.00018106	55.23031039	


Symbol 0
Price
1.012
1.01
1.008
1.006
1.004
1.002
09:40:03
09:55:03
10:10:03
10:25:03
10:40:03
10:55:03
11:10:03
13:15:06
13:30:06
13:45:06
14:00:06
14:15:06


Symbol 6
Price
1.11
1.108
1.106
1.104
1.102
1.1
1.098
1.096
09:40:03
09:55:03
10:10:03
10:25:03
10:40:03
10:55:03
11:10:03
13:15:06
13:30:06
13:45:06
14:00:06
14:15:06
Time



• 特征值构造

1）保留了示例代码中给出的几种构建方法：

①价格+1（n_ask1+1/n_bid1+1……）；

②价格差（ask1-bid1/ask2-bid2/ask3-bid3）；

③价量加权结合（ask1 与 n_bsize2 加权、ask2 与 n_ bsize1 加权…… ）

④买入与卖出量差（vo1_rel_diff 与 volall_rel_diff）

2）对成交金额变换：第一个想法是计算成交额/前一个交易日的 MAX 作为成交量的指标，但是受限于只能使用 100 个 tick 的数据，无法获取大部分前一个交易日的 MAX，所以对成交额的处理方式选择用“成交额/（1+n_midprice）”，然后取对数，作为成交量指标。

3）根据基础数据模拟日频数据计算 MACD bar 的值作为特征值。

4）模拟日频数据计算 KDJ 指标的 K 值、D 值和 J 值作为特征值。

5）计算 5 个 tick 的均值作为 5 日均线的模拟。

6）计算波动率：rate=(2+ask1+bid1)/(2+ask1befoe+bid1before)-1，before 分别取 5tick、10tick、20tick、40tick、60tick 前的值。

全部特征

```
0   date              int64
 1   time              object
 2   sym               int64
 3   n_close           float64
 4   amount_delta      float64
 5   n_midprice        float64
 6   n_bid1            float64
 7   n_bsize1          float64
 8   n_bid2            float64
 9   n_bsize2          float64
 10  n_bid3            float64
 11  n_bsize3          float64
 12  n_bid4            float64
 13  n_bsize4          float64
 14  n_bid5            float64
 15  n_bsize5          float64
 16  n_ask1            float64
 17  n_asize1          float64
 18  n_ask2            float64
 19  n_asize2          float64
 20  n_ask3            float64
 21  n_asize3          float64
 22  n_ask4            float64
 23  n_asize4          float64
 24  n_ask5            float64
 25  n_asize5          float64
 26  label_5           float64
 27  label_10          float64
```

```
 28  label_20          float64
 29  label_40          float64
 30  label_60          float64
 31  file              object
 32  hour              int64
 33  minute            int64
 34  spread1           float64
 35  spread2           float64
 36  spread3           float64
 37  mid_price1        float64
 38  mid_price2        float64
 39  mid_price3        float64
 40  weighted_ab1      float64
 41  weighted_ab2      float64
 42  weighted_ab3      float64
 43  vol1_rel_diff     float64
 44  volall_rel_diff   float64
 45  close             float64
 46  n_close_1         float64
 47  histogram         float64
 48  macd_values       float64
 49  signal_line       float64
 50  K                 float64
 51  D                 float64
 52  J                 float64
 53  processed_amount  float64
```

测试结果与参数调整

截至本报告写作时间，共成功提交 7 次模型，有 2 个模型尚未在公榜上跑出结果，下图

分别为测试结果、使用的特征值及模型参数调整。

	仿真平台															自测试														
	Label_5			Label_10			Label_20			Label_40			Label_60			Label_5			Label_10			Label_20			Label_40			Label_60		
提交序号	Precis	Recall	F_0.5	Precis	Recall	F_0.5	Precis	Recall	F_0.5	Precis	Recall	F_0.5	Precis	Recall	F_0.5	Precis	Recall	F_0.5	Precis	Recall	F_0.5	Precis	Recall	F_0.5	Precis	Recall	F_0.5	Precis	Recall	F_0.5
267	0.445	0.304	0.408	0.437	0.395	0.428	0.380	0.224	0.334	0.353	0.328	0.348	0.350	0.389	0.357	0.658	0.287	0.522	0.691	0.433	0.617	0.632	0.164	0.402	0.656	0.325	0.545	0.678	0.488	0.629
325	0.505	0.278	0.434	0.485	0.375	0.458	0.450	0.202	0.361	0.412	0.295	0.381	0.395	0.373	0.391	0.655	0.367	0.566	0.686	0.521	0.645	0.626	0.241	0.474	0.661	0.442	0.601	0.682	0.616	0.668
334	0.417	0.490	0.430	0.419	0.566	0.442	0.372	0.412	0.379	0.355	0.504	0.377	0.360	0.535	0.385	0.553	0.654	0.571	0.604	0.801	0.635	0.534	0.577	0.542	0.588	0.832	0.625	0.632	0.926	0.675
344	0.426	0.495	0.438	0.428	0.569	0.450	0.383	0.418	0.389	0.361	0.506	0.383	0.364	0.539	0.390	0.559	0.651	0.576	0.609	0.797	0.639	0.539	0.576	0.546	0.591	0.827	0.627	0.633	0.924	0.675
355	0.576	0.208	0.425	0.574	0.220	0.434	0.534	0.108	0.299	0.505	0.071	0.228	0.463	0.061	0.199	0.664	0.243	0.493	0.707	0.267	0.531	0.630	0.111	0.325	0.677	0.084	0.280	0.696	0.079	0.272
367																0.615	0.422	0.563	0.657	0.492	0.616	0.582	0.336	0.507	0.619	0.476	0.584	0.642	0.693	0.652
384																0.645	0.426	0.585	0.684	0.504	0.638	0.633	0.371	0.555	0.677	0.516	0.637	0.689	0.737	0.698

全部特征		训练序号						
序号	名称	267	325	334	344	355	367	384
1	sym		✓	✓	✓	✓	✓	✓
2	n_close	✓	✓	✓	✓	✓	✓	✓
3	amount_delta		✓	✓	✓	✓	✓	✓
4	n_midprice	✓	✓	✓	✓	✓	✓	✓
5	n_bid1	✓	✓	✓	✓	✓	✓	✓
6	n_bsize1	✓	✓	✓	✓	✓	✓	✓
7	n_bid2	✓	✓	✓	✓	✓	✓	✓
8	n_bsize2	✓	✓	✓	✓	✓	✓	✓
9	n_bid3	✓	✓	✓	✓	✓	✓	✓
10	n_bsize3	✓	✓	✓	✓	✓	✓	✓
11	n_bid4	✓	✓	✓	✓	✓	✓	✓
12	n_bsize4	✓	✓	✓	✓	✓	✓	✓
13	n_bid5	✓	✓	✓	✓	✓	✓	✓
14	n_bsize5	✓	✓	✓	✓	✓	✓	✓
15	n_ask1	✓	✓	✓	✓	✓	✓	✓
16	n_asize1	✓	✓	✓	✓	✓	✓	✓
17	n_ask2	✓	✓	✓	✓	✓	✓	✓
18	n_asize2	✓	✓	✓	✓	✓	✓	✓
19	n_ask3	✓	✓	✓	✓	✓	✓	✓
20	n_asize3	✓	✓	✓	✓	✓	✓	✓
21	n_ask4	✓	✓	✓	✓	✓	✓	✓
22	n_asize4	✓	✓	✓	✓	✓	✓	✓
23	n_ask5	✓	✓	✓	✓	✓	✓	✓
24	n_asize5	✓	✓	✓	✓	✓	✓	✓
25	hour	✓	✓	✓	✓	✓	✓	✓
26	minute	✓	✓	✓	✓	✓	✓	✓
27	spread1	✓	✓	✓	✓	✓	✓	✓
28	spread2	✓	✓	✓	✓	✓	✓	✓
29	spread3	✓	✓	✓	✓	✓	✓	✓
30	mid_price1	✓	✓	✓	✓	✓	✓	✓
31	mid_price2	✓	✓	✓	✓	✓	✓	✓
32	mid_price3	✓	✓	✓	✓	✓	✓	✓
33	weighted_ab1	✓	✓	✓	✓	✓	✓	✓
34	weighted_ab2	✓	✓	✓	✓	✓	✓	✓
35	weighted_ab3	✓	✓	✓	✓	✓	✓	✓
36	vol1_rel_diff	✓	✓	✓	✓	✓	✓	✓
37	volall_rel_diff	✓	✓	✓	✓	✓	✓	✓
38	n_close_1		✓	✓	✓	✓	✓	✓
39	histogram	✓	✓	✓	✓	✓	✓	✓
40	macd_values		✓	✓	✓	✓	✓	✓
41	signal_line		✓	✓	✓	✓	✓	✓
42	K		✓	✓	✓	✓	✓	✓
43	D		✓	✓	✓	✓	✓	✓
44	J		✓	✓	✓	✓	✓	✓
45	processed_amount	✓	✓	✓	✓	✓	✓	✓
46	rate_5				✓	✓	✓	✓
47	rate_10				✓	✓	✓	✓
48	rate_20				✓	✓	✓	✓
49	rate_40				✓	✓	✓	✓
50	rate_60				✓	✓	✓	✓
51	M5							✓

训练序号	主要变化
267	使用37个特征进行训练
325	特征增加到45个
334	特征不变，上涨下跌样本的权重*2，训练轮数增加
344	特征增加到50个
355	上涨下跌的概率要求 > 0.6
367	更加细化上涨下跌的判定
384	增加特征M5，训练轮数提升至800，加入过拟合后自动停止，使用交叉验证选择最优模型

训练参数		训练序号						
序号	名称	267	325	334	344	355	367	384
1	objective	multi:softprob						
2	num_class	3						
3	learning_rate	0.05						
4	max_depth	8		6				
5	subsample	0.8						
6	colsample_bytree	0.8						
7	seed	2024						
8	gamma	1						
9	min_child_weight	3		1				
10	lambda	10						
11	alpha	2						
12	tree_method	hist						
13	eval_metric	mlogloss						
14	num_boost_round	270		500				800
15	sample_weight	-	-	(2,1,2)				

• 参数调整

模型用到的参数有：

1）objective：这个参数用于定义优化算法的目标。不同的目标函数可以导致不同的优化结果和性能表现。

2）num_class：这个参数表示分类器的数量，通常在多类分类问题中使用。每个分类器负责识别一个类别，通过组合这些分类器的输出，实现对多个类别的预测。

3）learning_rate：学习率是优化算法中的一个重要参数，它决定了梯度下降法中每次迭代更新的步长。较高的学习率可能导致收敛过快而错过最优解，而较低的学习率则可能导致收敛速度太慢。

4）max_depth：最大深度限制了决策树或神经网络等模型的复杂度，从而防止过拟合。较小的最大深度意味着模型更简单，可能具有更好的泛化能力，但可能会牺牲一些准确性。

5）subsample：子采样比例是指在构建树的过程中，随机采样的数据占总数据的比例。设置为 0.8 意味着将随机选取 80%的数据来构建每棵树，这有助于减少方差，提高模型的稳定性。

6）colsample_bytree：这个参数与’subsample’类似，但它控制的是每棵树的特征采样比例。设置为 0.8 意味着在构建每棵树时，只使用 80%的特征，这样可以防止过度拟合，提高模型的泛化能力。

7）seed：种子数用来设置随机数的初始值，使得在不同的运行中获得相同的随机序列。这对于重复实验和比较不同方法的结果非常有用。

8）gamma：Gamma 参数是在 XGBoost 等梯度提升框架中使用的正则化项，它可以有助于避免过拟合。增加 Gamma 值通常会得到更简单的模型，但也可能降低准确性。

9）min_child_weight：这个参数定义了决策树分裂的最小样本权重之和。设置较高的值可以确保只有当有足够多的数据进行分裂时才进行分裂，从而防止模型过于复杂。

10）lambda（Lambda）：在某些上下文中，Lambda 可能指的是 L2 正则化系数，用于惩罚模型的复杂性，帮助预防过拟合。

11）alpha（Alpha）：在一些情况下，Alpha 可能代表 L1 正则化系数，也称为 Lasso 正则化，它通过添加绝对值损失来鼓励稀疏性，即产生更简单的模型。

12）tree_method：树方法参数选择不同的算法来实现决策树或随机森林等模型的训练过程。不同的实现可能会有不同的效率和效果。

13）eval_metric：评估指标是用来衡量模型性能的标准。mlogloss 通常指的是多类对数损失，它是多类分类问题中常用的评估指标之一。

14）num_boost_round：这个参数指定了梯度提升机中迭代的次数，也就是要构建多少个弱学习者（如决策树）并进行加权组合。

15）sample_weight：样本权重允许为训练集中的每个样本分配一个权重，以指示其在模型训练中的重要性。较大的权重意味着该样本对模型的影响更大，这在处理不平衡数据集时特别有用。

比较各个版本的不同，后面几个版本降低了 min_child_weight，适当增加了模型的复杂性，同时降低了 max_depth 以降低过拟合的可能性，另外增加了迭代次数，以提升训练效果。降低了预测值为 1 的权重，使得模型的盈利能力得以提升。355、367、384 分别对于模型判别标准进行了调整和细化，355 提高了判别标准，所以准确率明显提高，召回率大大降低，平均收益率大大提高，但是 f0.5 的值也降低了。367 取消了严格判别标准，改用概率差判别，召回率大大升高，f0.5 的值也有所提高，但总的来说各方面表现都比较平庸。384 增加了一个特征值，提升训练轮数到 800，并且使用了交叉验证，使得各个指标相对于 367 有全面提升，是目前自测试效果最好的。

改进方向讨论

准确率和召回率不可兼得，二者关乎平均收益与总收益，在现实交易中，每次交易都有摩擦成本，那么我们应该更看重哪个指标，如何把交易成本纳入模型？本次模型训练让我感受到了调整不同参数和规则如何影响模型的各项指标，为今后不同交易策略的量化工具的开发和选择提供了思路，作为开启量化道路之门的一把钥匙。