股票交易序列建模与预测大作业

—— 模拟股票期货交易策略与评估

负责助教：胡天成、朱轩宇

数据集描述

- 包含三十一支股票数据，包括股票每天的开盘收盘价，最高价和最低价。

Date	Open	High	Close	Low	Name
2006-01-03	61.07	61.85	61.05	61.63	JNJ

实际数据集会以股票名称区分，并以.csv的格式发出，例如JNJ.csv

作业环境

- 基本环境：
  - python==3.9
  - numpy
  - pandas
  - jupyter
  - matplotlib
  - scikit-learn
  - PIL
  - opencv-python
  - img2vec-pytorch
  - torch
  - torchvision
- 如果你要使用其他库，请在notebook里加入%pip install …方便助教运行

内容概览（共100分）

- 股票交易序列的基础特征提取与可视化（10分）
- 基础交易策略的实现与评估（20分）
- 基于交易信息的股票聚类与类型化（20分）
- 对交易数据进行时间序列建模和回归，完成预测模型。结合上面的聚类信息，特征信息等进行优化，评估（40分）
- 对盈亏结果进行分析，给出合理的推断和假设，设想可能的改进方案（10分）

1.1、股票的基础特征提取与直观对比（10分）

根据给出的股票交易数据，对每支股票实现基础交易量化指标的计算。包括：收益率均值、波动率、夏普比率，最大回撤、偏度与峰度。（6分）

针对不少于3个指标结合个股进行分析。选择多只有代表性的股票进行对比可视化，反映对相关指标的定性表达（4分）

1.1、股票的基础特征提取与直观对比（续）

1. 收益率均值

定义：一段时间内每日收益率的算术平均值

意义：反映资产的平均盈利能力，是评估投资表现的基准指标

2. 收益率波动率

定义：收益率的标准差

意义：衡量资产价格波动幅度，数值越大代表风险越高

3. 夏普比率

定义：(收益率均值 - 无风险利率)/波动率

意义：衡量单位风险获得的超额收益， > 2为优秀水平

1.1、股票的基础特征提取与直观对比（续）

4. 最大回撤

定义：选定周期内从峰值到谷底的最大损失幅度

意义：反映极端风险下的潜在亏损，数值越小抗风险能力越强

5. 偏度

定义：收益率分布的不对称程度

意义：正偏态预示暴涨概率 > 暴跌概率

6. 峰度

定义：收益率分布的尖峰程度

意义： > 3表示极端行情概率高于正态分布

注：部分指标（如夏普比率、最大回撤）的计算方法存在多种变体，实际应用中需注意参数。
此外DIS.csv中存在着空item，需要有对应的错误处理方式，比如将空值替换为相邻数据的均值。
指标的计算与可视化可以合理利用AI大模型的支持与帮助。

1.2、股票的高级指标的合理运用（0分）

1. Hurst指数

定义：衡量时间序列均值回归特性的指标

意义：＞0.5趋势延续，＜0.5均值回归

2. MACD均值

定义：移动平均收敛发散指标的均值

意义：反映中长期趋势强度

3. RSI_14均值

定义：14日相对强弱指标均值

意义：30-70区间外预示超买超卖

4. 布林带穿透率

定义：价格突破布林带上/下轨的频率

意义：衡量市场极端波动概率

合理选取不少于2个高级指标加入自己的策略模型，并在报告中详细解释这些指标的选取思路及实用效果（前后对比？）

1.2、股票的高级指标的合理运用（续）

5. 量价相关系数

定义：成交量与价格的相关系数

意义：反映资金推动效应强弱

6. 隔夜跳空概率

定义：开盘价与前收盘价差距 > 1%的概率

意义：衡量市场隔夜信息冲击强度

7. 周内效应强度

定义：各交易日收益率的标准差

意义：捕捉日历效应中的规律性波动

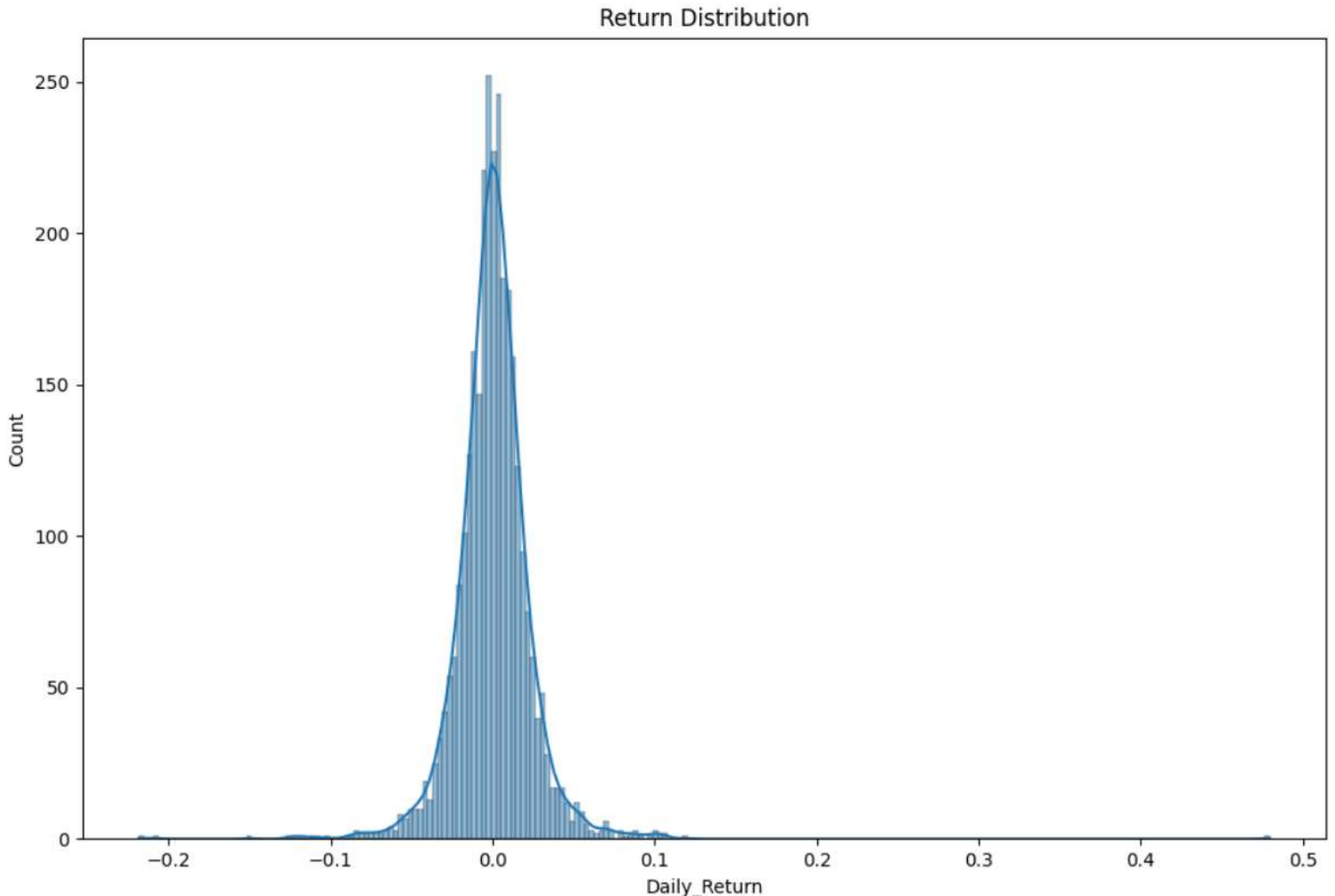
8. 月波动聚集性

定义：月度波动率的自相关性

意义：反映风险传染特征， > 0.5存在波动聚集

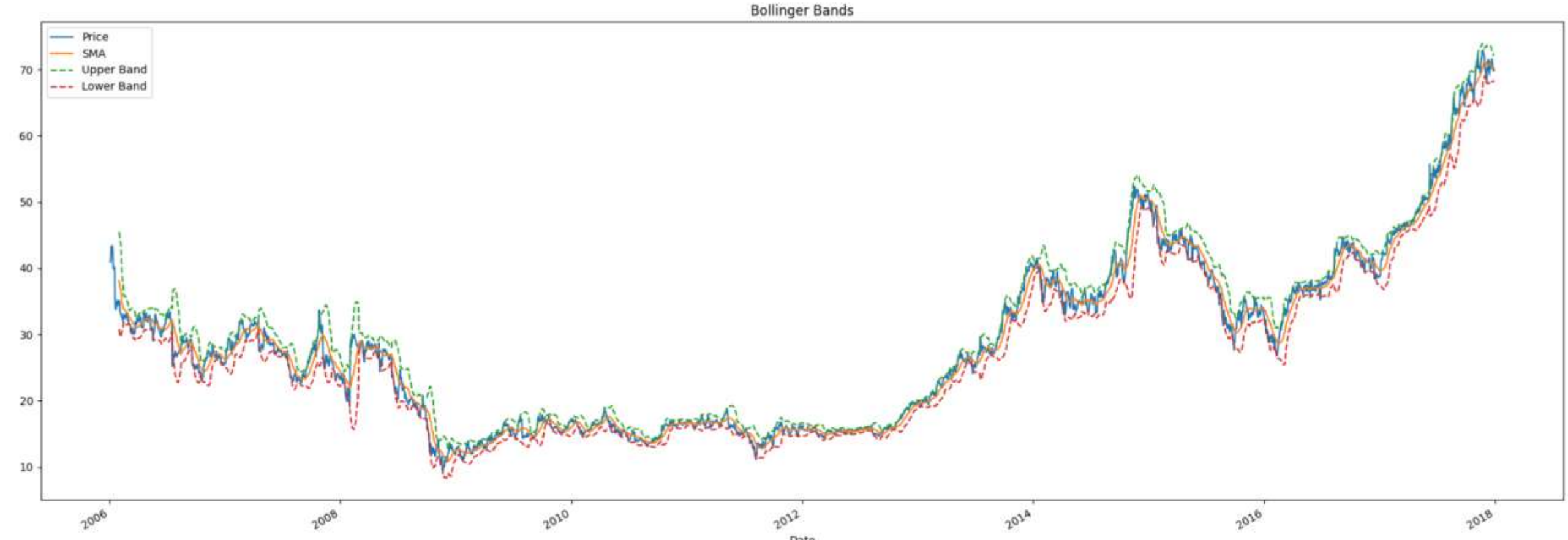
股票特征提取与可视化样例-1

日利率分布图

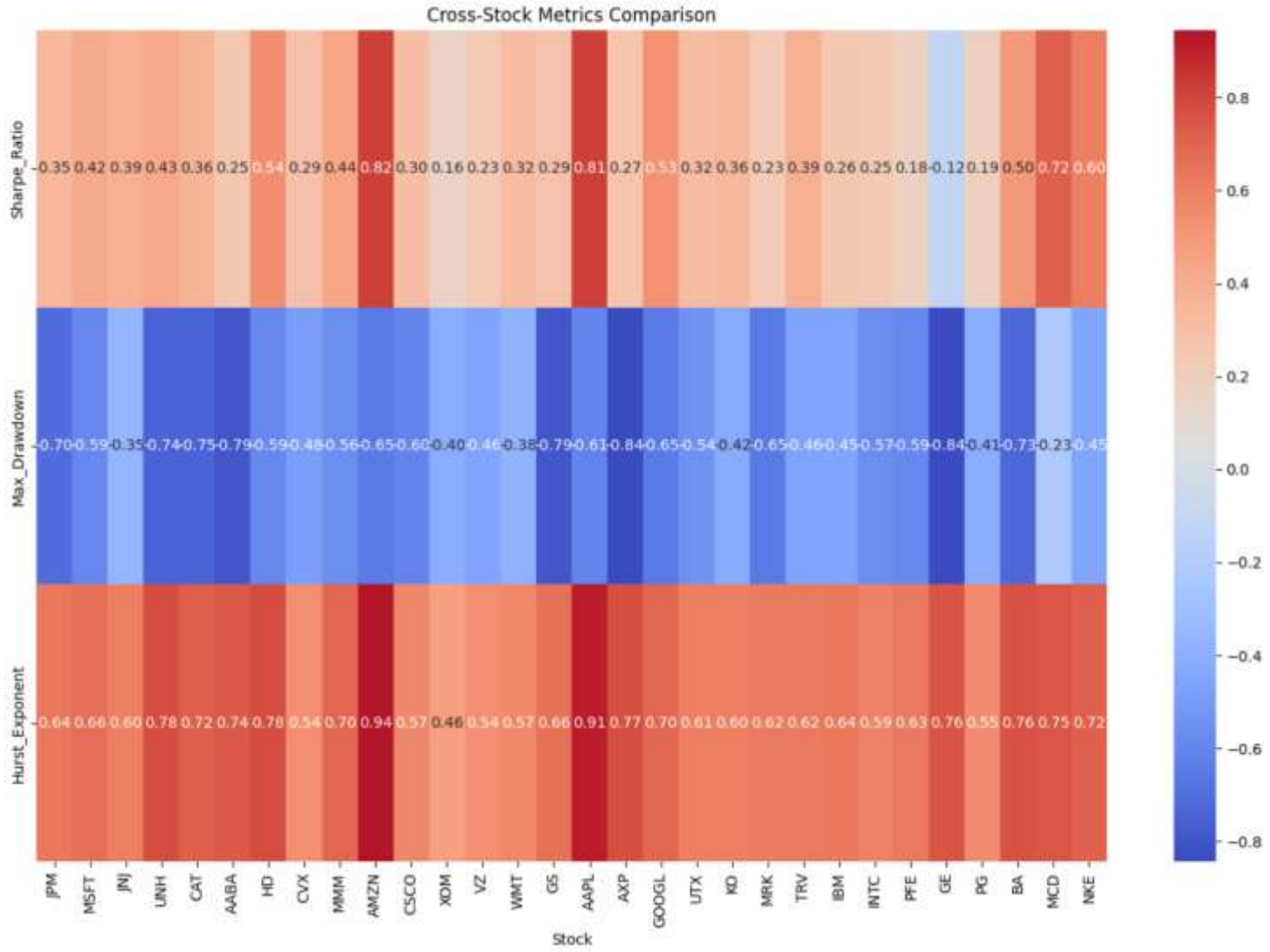


股票特征提取与可视化样例-2

布林带



股票特征提取与可视化样例-3



股票特征横向比较
给出分析（？）

二、基础交易策略的实现与评估（20分）

1、从给定的五大种基础交易策略中任选3种具体的交易策略进行实现（15分）
对于每种策略:

1. 成功实现完整策略代码（2分）
2. 可视化数据图 （1分）
3. 对结果进行合理的分析，指出策略的适用场景或失败因素。结合一些静态动态指标，提出合理的改进方案并对比效果(2分)

2、设计一个组合策略，可以加入静态动态的特征，综合利用多种交易策略，实现好的交易成绩。并同样完成可视化及分析。（5分）

二、基础交易策略简介

策略分类及基本代码

一、趋势跟踪策略

均线交叉策略，原理：短期均线上穿长期均线时买入，下穿时卖出

```
# 双均线策略实现
df['SMA_10'] = df['Close'].rolling(10).mean()  # 10日均线
df['SMA_30'] = df['Close'].rolling(30).mean()  # 30日均线
df['Signal'] = np.where(df['SMA_10'] > df['SMA_30'], 1, -1)
```

MACD策略，原理：DIF线上穿DEA线时买入，反之下穿卖出

```
# MACD指标计算
df['EMA_12'] = df['Close'].ewm(span=12).mean()
df['EMA_26'] = df['Close'].ewm(span=26).mean()
df['DIF'] = df['EMA_12'] - df['EMA_26']
df['DEA'] = df['DIF'].ewm(span=9).mean()
df['Signal'] = np.where(df['DIF'] > df['DEA'], 1, -1)
```

二、基础交易简介（续）

布林带策略，原理：价格触及下轨买入，触及上轨卖出

```
df['MA_20'] = df['Close'].rolling(20).mean()
df['Upper'] = df['MA_20'] + 2*df['Close'].rolling(20).std()
df['Lower'] = df['MA_20'] - 2*df['Close'].rolling(20).std()
df['Signal'] = np.where(df['Close'] < df['Lower'], 1,
                        np.where(df['Close'] > df['Upper'], -1, 0))
```

RSI超买超卖策略，原理：RSI低于30买入，高于70卖出

```
delta = df['Close'].diff()
gain = delta.where(delta > 0, 0)
loss = -delta.where(delta < 0, 0)
avg_gain = gain.rolling(14).mean()
avg_loss = loss.rolling(14).mean()
rs = avg_gain / avg_loss
df['RSI'] = 100 - (100 / (1 + rs))
df['Signal'] = np.where(df['RSI'] < 30, 1,
                        np.where(df['RSI'] > 70, -1, 0))
```

二、基础交易策略简介（续）

三、统计套利策略

配对交易策略，原理：两只相关性强的股票价差偏离均值时做多低估/做空高估

```
spread = df['StockA'] - df['StockB']
zscore = (spread - spread.rolling(30).mean()) / spread.rolling(30).std()
df['Signal'] = np.where(zscore < -1.5, 1,
                        np.where(zscore > 1.5, -1, 0))
```

四、动量策略

动量轮动策略，原理：选择过去N日收益率最高的标的持有

```
df['Momentum'] = df['Close'].pct_change(20)
# 选择动量前3的股票
top_stocks = df.groupby('Date')['Momentum'].nlargest(3).index.get_level_values(1)
```

五、复合策略示例

海龟交易法则（含仓位管理）

```
df['TR'] = np.maximum(df['High']-df['Low'],
                      np.maximum(abs(df['High']-df['Close'].shift()),
```

二、基础交易策略的实现与评估（示例）

以双均线策略为例：

代码见附件

我们对于AABA股票采取双均线交易策略可以得到如下结果：

总收益 (Total Return = –81.72%)

策略期间累计跌去了超过 80% 的本金，意味着如果一开始投入 1 万元，到最后只剩下不到 2 千元。

年化收益率 (CAGR = –13.22%/年)

每年平均亏损13%，长期来看不仅没跑赢大盘，甚至大幅跑输“买入并持有”策略（AABA 在这段时间内虽有波动，但整体没亏得如此惨烈）。

年化波动率 (Annual Volatility ≈ 38.4%)

策略返回标准差接近 38%，波动极大。在收益本来就是负数的情况下，这么高的波动性反而让人更难承受。

夏普比率 (Sharpe Ratio ≈ –0.16)

负值表明策略单位风险承担带来了负收益——换言之，承担风险还不如拿现金不动。通常认为 Sharpe Ratio 低于 0.5 就很难接受，这里更是惨烈地跑到负值。

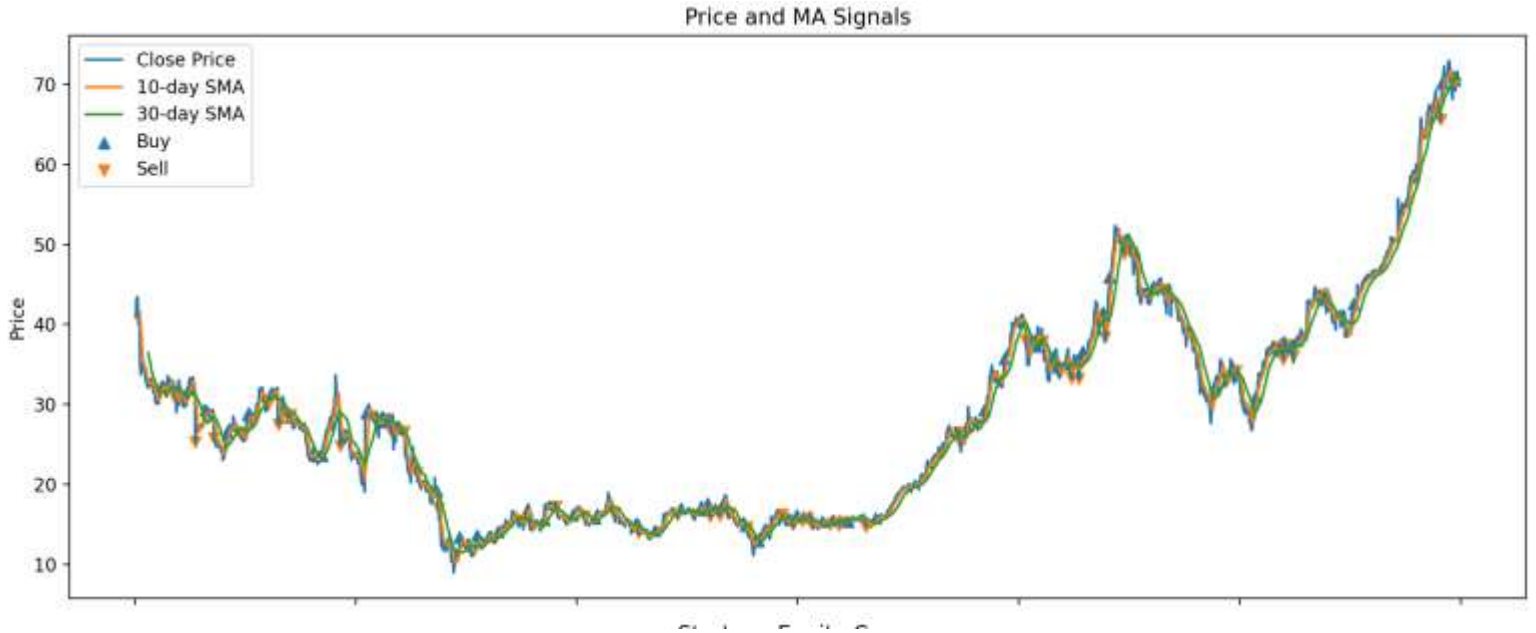
最大回撤 (Max Drawdown ≈ –92.06%)

最大回撤超过 90%，说明策略中曾经出现过接近清仓的亏损峰值——几乎所有资金在某些时点都被“一扫而空”。

二、基础交易策略的实现与评估（示例）

可视化分析:

这张图帮助我们直观地看到在价格上涨趋势中“金叉”是否及时捕捉了买入机会，以及在下跌趋势中“死叉”是否避免了更深的回撤。若信号多出现在震荡或逆势中，就可能造成虚假交易。

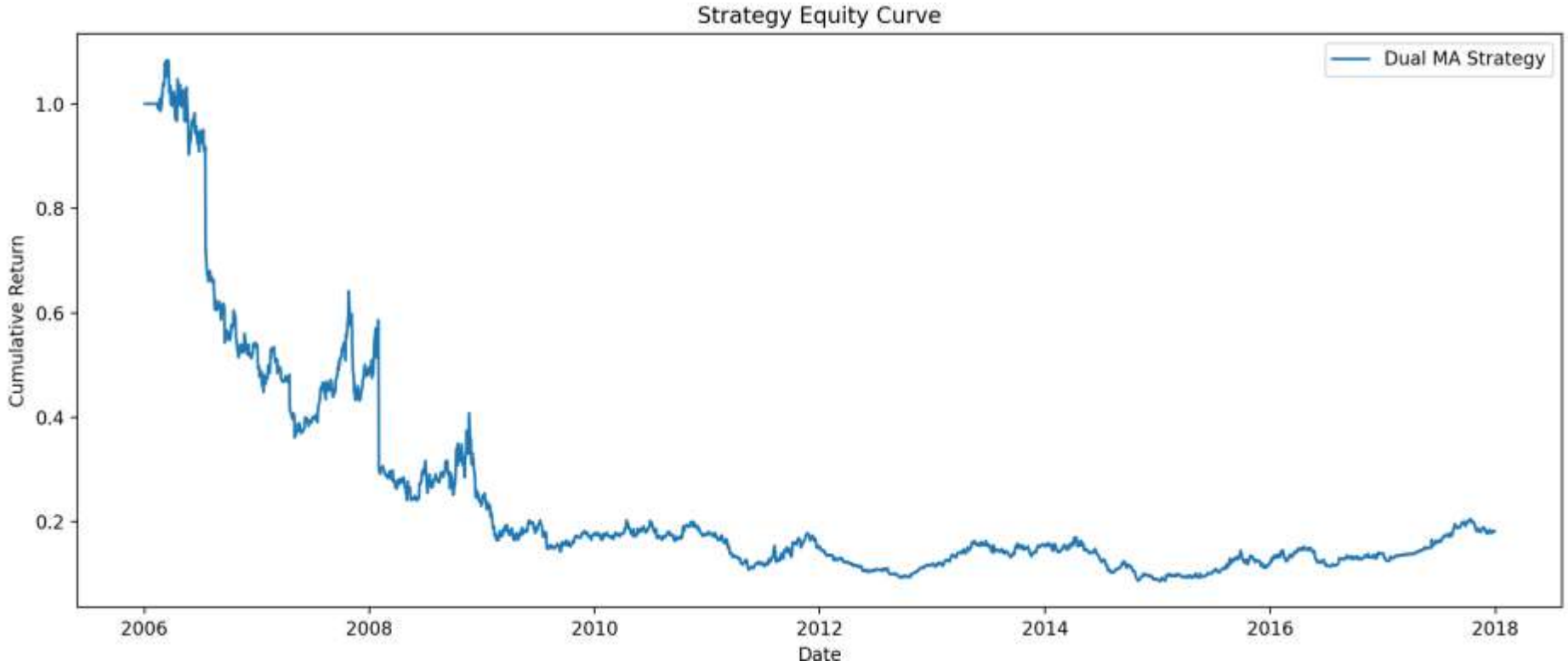


二、基础交易策略的实现与评估（示例）

（策略权益曲线）图

•曲线的陡峭程度体现短期获利或亏损的速度；平缓波动则表示策略在震荡市中的持仓来回切换。

•若出现大幅回撤或长时间下行，说明策略在大趋势下伤害严重，风险控制不足。



二、基础交易策略的实现与评估（示例）

原因分析：AABA 股价长期下行，趋势跟踪策略在下跌趋势里会不断锁定亏损。10/30 均线窗口对快速波动响应不足，又在平缓震荡期频繁交叉、产生“虚假信号”。

注意，均线分析很容易遇到均值化之后数据不对齐等问题，比如：
移动平均线（MA20/MA50）的计算会生成前20/50天的NaN值，导致数据长度缩短

三、基于交易信息的股票聚类与类型化（20分）

采用不少于两种聚类方法实现股票聚类及可视化分析
1、利用股票的静态特征进行聚类：如kmeans等
2、利用股票交易的动态特征进行聚类：如滚动相关性计算+社区划分
3、截取特定时间窗口进行相关性分析或通过滑动窗口动态对齐（DTW）捕捉局部模式的聚类方案

聚类方法实现（10分）：
可视化及类型化分析（给出聚类结果的合理解释？）（5分）
根据目标任务对聚类方案进行优化改进（5分）

三、以Kmeans为例给出一个参考样例:

1. 金融特征提取
   分别提取出每支股票的一些特征指标例如：总回报，复合年增长率，波动率，夏普，最大回撤等等;
2. 分析股票背后的公司特征
   可以利用yifinance根据股票名（AABA）去查询背后的公司并获取公司的一些特征，比如公司类型;

Company Information:

ticker	longName	sector	industry	marketCap
JPM	JPMorgan Chase & Co.	Financial Services	Banks - Diversified	6.798796e+11
MSFT	Microsoft Corporation	Technology	Software - Infrastructure	2.913005e+12
JNJ	Johnson & Johnson	Healthcare	Drug Manufacturers - General	3.727728e+11
UNH	UnitedHealth Group Incorporated	Healthcare	Healthcare Plans	3.861623e+11
CAT	Caterpillar Inc.	Industrials	Farm & Heavy Construction Machinery	1.466581e+11

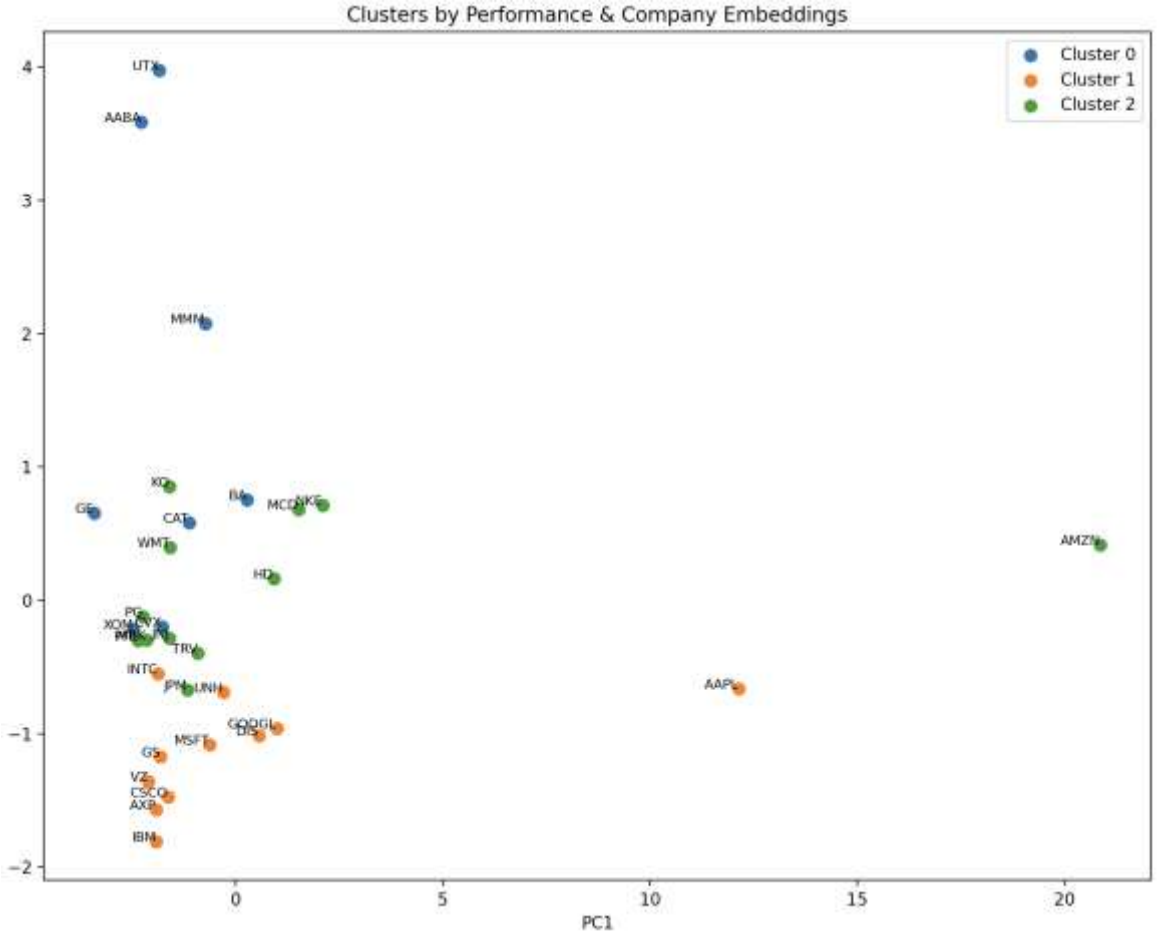
三、以Kmeans为例的一个参考样例：（续）

3. 使用小维度词向量（Glove）将文本（公司类型）编码为词向量并将其与金融效益综合考虑进行聚类

```
text_features = (df['sector'] + ' ' + df['industry']).fillna('')
embed_vecs = np.vstack([text_to_embedding(t, embeddings, dim) for t in text_features])
embed_df = pd.DataFrame(embed_vecs, index=df.index, columns=[f'embed_{i}' for i in range(dim)])
```

三、以Kmeans为例的一个参考样例：（续）

4. 确定聚类数k，并进行聚类，画出可视化图，并进行性能评估



Silhouette Score仅有0.1123，kmeans聚类效果非常差，请思考优化方案

四、结合上面的聚类信息，特征信息，针对交易数据进行时间序列建模和回归，完成预测模型及优化，评估（40分）

决策生成函数：

```
def generate_strategy(self, portfolio, date, real_value, next_trading_date=None):
    """
    生成每日交易策略，根据当前日期和下一个交易日的日期间隔调整策略
    :param
    portfolio: 当前投资组合字典，包含以下字段：
    {
        'cash': 当前现金余额 float,
        'holdings': 目前持股信息 {stock: shares},
        'transaction_log': 历史交易记录 []
    }
    date: 需要决策的日期k->str
    real_value: 包含前面k-1天股票的真实开盘收盘价，最高价和最低价->Dict[str, List[List[Any]]]
    {
        'AAPL':[[Date,Open,High,Low,Close,Volume,Name], [Date,Open,High,Low,Close,Volume,Name], ......]
    }
    next_trading_date: 下一个交易日的日期，决定是短期还是长期策略
    :return: 交易策略列表->List[Dict[str, Dict[str, Any]]]：
    [
        {'AAPL': {'action1': 'buy', 'shares1': 100, 'action2':'sell', 'shares2':50}},
        {'MSFT': {'action1': 'none', 'shares1': 0, 'action2': 'none', 'shares2': 50}}
    ]
    """
```

四、对交易数据进行时间序列建模和回归，完成预测模型。结合上面的聚类信息，特征信息等进行优化，评估（40分）

约束：

1. 给定未来两个测试点的时期，需要决策第一个测试点的买入卖出，并在第二个测试点前一个交易日平仓，因此每次输入进行决策的时候手里都是空仓，但是现金余额会不断变化
2. 限制最多对六只股票进行九种操作，即每一只股票可以有两个action，每个action可以有三种操作：buy, sell, none，当action为buy或者sell时需要指定后续shares为一个正整数
3. 风控参数：
    1. max_shares_per_trade = 10000：限制单次交易最大股数为1万股
    2. max_position_value_ratio = 0.2：限制单个股票持仓价值不超过总资产的20%
    3. max_short_ratio = 0.5：限制最大卖空比例为总资产的50%

五、对盈亏结果进行分析，给出合理的推断和假设，设想可能的改进方案（10分）

最后请摘出与作业要求对应的关键代码和结果，标出对应题号并简要概括代码，写一个方便助教理解并对照要求给分的实验报告

作业提交截止时间：2025年5月11号 晚上23点

提交方式：例如：Q1.ipynb文件，Q2.ipynb, ……，以及作业的实验报告

选择用 学号.**zip**压缩提交（扩展名是.zip）。