参考股票软件的相关设计模式

历史价格数据，比如开盘价、收盘价、最高价、最低价、成交量等基本特征。不过，为了预测未来趋势，可能需要生成更多的相关特征。比如技术指标，像移动平均线、MACD、RSI这些，都是常用的。不过需要考虑不同时间窗口的选择，比如7天、14天、30天的均线，可能捕捉到不同时间尺度的趋势。

可能需要处理滞后特征，也就是过去几天的数据作为特征。比如用前1天、前3天、前5天的收盘价作为输入，帮助模型捕捉历史模式。但要注意避免过拟合，可能需要交叉验证来确定最佳的滞后步长。

接下来，波动性相关的特征也很重要。比如计算每日价格变化的波动率，或者使用ATR（平均真实波幅）来衡量市场波动情况。波动率大的时候可能预示着趋势的变化，这可能对分类模型有帮助。

另外，需要考虑时间相关的特征，比如星期几、月份、季度等。股市可能有季节性效应，比如“一月效应”或者周末效应。将这些作为类别特征，或许能帮助模型识别时间模式。

还有，外部数据源的引入可能有用。比如经济指标（GDP、失业率）、利率、其他指数（如S&P500，NASDAQ）的表现，甚至新闻情绪分析。不过这可能涉及到数据获取和整合的问题

使用Pearson相关系数、互信息法，或者基于模型的特征重要性（如随机森林、XGBoost的特征重要性评分）。这样可以减少维度，防止过拟合，提高模型效率。

处理缺失值也是关键。比如前向填充，或者删除缺失的数据点。技术指标如MACD可能需要足够的数据才能计算，所以数据起始部分可能会有缺失，需要处理。

目标变量的处理。因为是预测下周的涨跌，可能需要将收盘价转化为分类标签，比如下周收盘价比本周高则为1，否则为0。或者考虑多分类，比如大涨、小涨、持平、小跌、大跌。但需要确保数据分布平衡，否则可能需要重采样方法。

* 构造5/10/20/60日等不同周期的移动平均线（MA），计算价格与MA的偏离度：

df['MA5'] = df['Close'].rolling(5).mean()

df['Price\_MA5\_Deviation'] = (df['Close'] - df['MA5']) / df['MA5']

* 计算ATR真实波动率(14日窗口)：

high\_low = df['High'] - df['Low']

high\_close = np.abs(df['High'] - df['Close'].shift())

low\_close = np.abs(df['Low'] - df['Close'].shift())

tr = pd.concat([high\_low, high\_close, low\_close], axis=1).max(axis=1)

df['ATR'] = tr.rolling(14).mean()

* 复合动量因子（3日动量与10日动量的相互作用）：

df['Momentum3'] = df['Close'].pct\_change(3)

df['Momentum10'] = df['Close'].pct\_change(10)

df['Momentum\_Ratio'] = df['Momentum3'] / (df['Momentum10'] + 1e-6)

* 构造Put/Call Ratio的5日EMA：

df['PCR\_EMA5'] = df['PutCallRatio'].ewm(span=5).mean()

* 构建VIX恐慌指数的趋势变化：

df['VIX\_Change'] = df['VIX'].pct\_change(3)

- 成交量异动指标（3日成交量Z-Score）：

df['Volume\_Z3'] = (df['Volume'] - df['Volume'].rolling(3).mean()) / df['Volume'].rolling(3).std()

傅里叶变换提取周期特征​

close\_fft = np.fft.fft(df['Close'].values)

freqs = np.fft.fftfreq(len(close\_fft))

df['Dominant\_Freq'] = freqs[np.argmax(np.abs(close\_fft))]

波动率结构分解​

garch = arch\_model(df['Returns'], vol='GARCH', p=1, q=1)

res = garch.fit()

df['Conditional\_Volatility'] = res.conditional\_volatility

市场状态编码​

hmm\_model = GaussianHMM(n\_components=3).fit(df[['Returns','Volume']])

df['Market\_State'] = hmm\_model.predict(df[['Returns','Volume']])

三、特征优化方法

1. ​​动态特征选择​

selector = FeatureUnion([

('variance', VarianceThreshold(threshold=0.1)),

('mutual', SelectKBest(mutual\_info\_classif, k=20))

])

optimized\_features = selector.fit\_transform(X, y)

1. 经济周期敏感度调整​

df['IP\_Vol\_Interaction'] = df['IndustrialProduction'] \* df['ATR']

## 四、特征工程注意事项

​​避免前瞻偏差​​

所有滚动计算必须采用严格滞后窗口：

df['MA5'] = df['Close'].shift(1).rolling(5).mean() # 确保无未来信息

​​市场机制过滤​​

在熔断交易日（如2020-03-16）添加机制标识特征：

df['CircuitBreaker'] = (df['DailyRange'] > 0.07).astype(int)

​​流动性调整​​

df['Illiquidity'] = (df['Close'].diff().abs() / df['Volume']).rolling(5).mean()

验证策略

建议采用经济周期敏感的回测方法：

from sklearn.model\_selection import TimeSeriesSplit

tscv = TimeSeriesSplit(n\_splits=5, test\_size=22) # 模拟月度调仓周期

for train\_index, test\_index in tscv.split(X):

X\_train, X\_test = X.iloc[train\_index], X.iloc[test\_index]

y\_train, y\_test = y.iloc[train\_index], y.iloc[test\_index]

特征效果验证

使用SHAP值进行经济解释：

explainer = shap.TreeExplainer(model)

shap\_values = explainer.shap\_values(X\_test)

shap.summary\_plot(shap\_values, X\_test, plot\_type="bar")

最终特征集应包含30-50个核心特征，在LightGBM模型中建议采用以下超参数配置：

params = {

'objective': 'binary',

'learning\_rate': 0.01,

'num\_leaves': 31,

'max\_depth': 5,

'min\_child\_samples': 100,

'subsample': 0.8,

'colsample\_bytree': 0.7,

'reg\_alpha': 0.1,

'reg\_lambda': 0.1

}

在模型监控中设置特征稳定性指标（PSI）阈值<0.1，当市场波动率（VIX>30）时自动触发特征权重再平衡机制。