

J-Quants APIを用いた日本株寄り引けロングショート戦略 検証計画

ご提示いただいた3つのドキュメントに基づき、J-Quants APIを用いた日本株寄り引けロングショート戦略の検証を行うための計画をご提案します。この計画は、ベースライン戦略の評価から始め、提案されている改善策(特にTabPFNの導入)を段階的に検証し、最終的に戦略の実用性を評価することを目的とします。

現在日時: 2025年5月8日

前提:

- J-Quants APIの利用アカウント及び必要なプラン(データ取得期間や種類に応じて)が準備されていること。
- Pythonを中心とした開発環境が構築可能であること。

第1フェーズ: ベースライン戦略の再現と評価

目的: ドキュメント1「ゼロから始める株式分析: J-Quants APIを用いた日本株寄り引けロングショート戦略分析_コード」で説明されているLightGBMを用いた基本戦略の性能を再現し、評価の基準点を確立します。

期間: 1~2週間

ステップ:

1. 環境構築:
 - Python環境設定 (例: Python 3.9+)。
 - 必要なライブラリのインストール: jquntsapi, pandas, numpy, scikit-learn, lightgbm, matplotlib, seaborn, shap。
2. データ収集と準備:
 - J-Quants APIを利用して、ドキュメント1で指定されている期間(例: 過去5年分、訓練3年・テスト2年)のTOPIX500構成銘柄の日足OHLCVデータを取得。
 - データクレンジング、欠損値処理を実施。
3. 特徴量エンジニアリング:
 - ドキュメント1に記載されている特徴量(過去リターン、移動平均乖離率、ボラティリティなど)を計算。
 - 目的変数(当日寄り付きから引けまでのリターン)を定義。
 - 必要に応じて特徴量のビンニング処理(KBinsDiscretizer)を適用。
4. モデル構築と学習:
 - LightGBM回帰モデルを構築。
 - 訓練データを用いてモデルを学習。
5. モデル評価と解釈 (初期):
 - テストデータを用いてモデルの予測性能を評価。
 - 評価指標: 日毎の予測値と実測値の相関係数(平均、中央値、標準偏差)、シャープレシオ、日毎の正答率。
 - shapライブラリを用いて特徴量の重要度を分析。
 - ドキュメント1に記載されている結果(例: 正答率約53%)との比較。

第2フェーズ: TabPFNモデルの導入と評価

目的: ドキュメント3「J-QuantsとTabPFN戦略応用検討」に基づき、TabPFNを導入し、ベースライン

戦略との性能比較を行います。

期間: 2~4週間

ステップ:

1. 環境設定:
 - tabpfn (または tabpfn-time-series) ライブラリ及びその依存関係(PyTorchなど)をインストール。
 - GPU環境の利用を検討(TabPFNの推論速度向上のため)。
2. データ準備 (TabPFN向け):
 - 各銘柄の時系列データをTabPFNの入力形式(表形式)に変換。
 - TabPFNの入力制約(サンプル数、特徴量数)を確認し、必要に応じてローリングウィンドウアプローチを設計・実装。
 - 例: ドキュメント3で言及されている最大1000~10000サンプル、最大100~500特徴量。
3. 特徴量エンジニアリング (TabPFN向け):
 - tabpfn-time-seriesで推奨されるタイムスタンプ由来の特徴量(カレンダー特徴、周期的エンコーディング、実行インデックス)を生成。
 - ベースライン戦略で有効だった金融特徴量(前日までのリターン、ボラティリティなど)を外生変数として追加。データリークageが発生しないよう注意。
4. モデル呼び出しと予測生成:
 - 各銘柄、各予測日に対してTabPFNモデルを呼び出し、翌日の寄り引けリターンを予測。
 - TabPFNの「コンテキスト内学習」の特性を理解し、適切な履歴データを入力として使用。
5. バックテスト (TabPFN):
 - 生成された予測シグナルに基づき、ロング・ショートポートフォリオを構築。
 - ベースライン戦略と同様の評価指標でバックテストを実施。
6. 性能比較と解釈:
 - LightGBMモデルとTabPFNモデルの性能を比較(シャープレシオ、累積リターン、最大ドローダウンなど)。
 - shap等を用いてTabPFNの予測根拠を分析(ドキュメント3の提案に従う)。
 - TabPFNの学習速度、実装の容易さなども評価。

第3フェーズ: バックテストの厳密化と高度な分析

目的: ドキュメント2「ゼロから始める株式分析: J-Quants APIを用いた日本株寄り引けロングショート戦略分析_ブラッシュアップ案」で提案されている、より現実的で信頼性の高いバックテスト手法を導入し、戦略の堅牢性を評価します。

期間: 2~3週間

ステップ:

1. バックテストライブラリの導入検討:
 - VectorBT, Backtesting.py, Backtraderなどの高機能バックテストライブラリのいずれかを選択し、導入。
2. ウォークフォワード分析の実装:
 - 訓練期間とテスト期間を時間軸に沿ってスライドさせながら繰り返し評価を行い、モデルの安定性やパラメータの頑健性を検証。
3. 時系列クロスバリデーションの導入:
 - Purged K-Foldクロスバリデーションなど、金融時系列データ特有の自己相関や情報リークageを考慮した手法を実装。

4. 取引コストとスリッページの考慮:
 - 想定される取引手数料、スプレッドによるスリッページをバックテストに組み込み、より現実的なパフォーマンスを評価。
 - pyfolioなどのライブラリを活用し、詳細なパフォーマンス分析(ティアシート生成など)を実施。
5. 感度分析:
 - 主要なパラメータ(例: TabPFNのコンテキストウィンドウ長、特徴量セット、ポートフォリオ構築ルール)を変更した場合のパフォーマンスの変化を分析。

第4フェーズ (オプション): その他の改善案の検討と実験

目的: ドキュメント2で提案されているその他の改善策について、実現可能性と期待効果を考慮し、優先度の高いものから実験的に導入・評価します。

期間: 2~4週間 (選択する項目による)

ステップ (例):

1. **Triple Barrier Labeling (TBL)** の導入:
 - 目的変数の定義を寄り引けリターンからTBLに変更し、リスクリワードを考慮した戦略の学習を試みる。
 - mlfinlabなどのライブラリを活用。
2. **LSTM**モデルの検討:
 - 生のOHLCVデータから直接パターンを学習するLSTMモデルを構築し、性能を比較。TensorFlow (Keras) や PyTorch を使用。
 - 実装コストと学習時間を考慮。
3. 特徴量の追加・最適化:
 - J-Quants APIで提供される新しいデータ(例: 2025年5月までに信用取引残高の日次公表銘柄詳細などが追加されていれば)の活用を検討。
 - Optunaなどを用いたハイパーパラメータ最適化。
4. アンサンブル学習の検討:
 - LightGBMとTabPFN、あるいは他のモデルの予測を組み合わせるアンサンブル手法を試す。

第5フェーズ: 総合評価と報告

目的: 全ての検証結果を統合し、戦略の有効性、実用性、限界、そして今後の改善方向について総合的な評価を行います。

期間: 1週間

ステップ:

1. 結果の集約と分析:
 - 各フェーズで得られた評価指標、分析結果を整理。
 - 各モデル/アプローチの強み、弱み、計算コスト、実装の複雑さを比較。
2. 戦略の堅牢性評価:
 - 異なる市場環境(ブル相場、ベア相場、レンジ相場など)でのパフォーマンスの安定性を評価。
3. 最終報告書の作成:
 - 検証目的、手法、結果、考察、今後の課題と推奨事項をまとめた報告書を作成。
 - 最も有望な戦略とその運用上の注意点を明確にする。

共通事項:

- バージョン管理: Git等を用いてコードと実験設定を管理。

- ドキュメンテーション: 各ステップのコード、設定、結果、考察を詳細に記録。
- **J-Quants API**の利用規約遵守: 特に自動売買に関する規約を確認(ドキュメント3で言及)。
- 最新情報の確認: J-Quants APIの仕様変更、ライブラリのアップデートなどに注意。

この計画はあくまでたたき台であり、検証の進捗や得られた結果に応じて、柔軟に調整していくことが重要です。特に、TabPFNやLSTMのような比較的新しい、あるいは複雑な技術の導入には、予期せぬ課題が発生する可能性も考慮に入れる必要があります。