Jane Street Kaggleコンペティションからの有用な知見に関する分析レポート

I. 序論：Jane Streetチャレンジの航海

A. Jane Street Kaggleコンペティションの概観

Jane Streetが主催するKaggleコンペティションは、匿名化された高頻度市場データを用いて市場の動きを予測するという、非常に複雑でハイステークスな課題を提示しています 。これらのコンペティションの目的は、参加者が実際の取引システムから派生したデータを用いてモデルを構築し、現代の金融市場における取引の成功に伴う日々の課題を垣間見ることです 。主催者や参加者が指摘するように、これらの課題には固有の困難さが伴います。具体的には、低いシグナル対ノイズ比、非定常な時系列データ、ファットテール分布、匿名化された特徴量、そしてスキルと運を区別することの難しさなどが挙げられます 。
コンペティションは進化しており、2021年の「Market Prediction」ではユーティリティスコア（u = \min(\max(t, 0), 6) \sum p_i、ここで t = \frac{\sum p_i}{\sqrt{\sum p_i^2}} \sqrt{\frac{250}{|i|}}、p_i = \sum_j (weight_{ij} * resp_{ij} * action_{ij})）が評価指標でした 。一方、2024/25年の「Real-Time Market Data Forecasting」では、responder_6 に対するサンプル加重ゼロ平均R二乗スコア（R^2 = 1 - \frac{\sum w_i (y_i - \hat{y}_i)^2}{\sum w_i y_i^2}）が評価指標となり、予測フェーズが導入されました 。

B. レポートの目的と範囲

本レポートの目的は、提供された情報源（---）に限定して、これらのコンペティションから得られた実用的な洞察と戦略を統合することです。これには、勝利したアプローチ、一般的な落とし穴、そして同様の定量的金融課題に取り組むための主要な学びが含まれます。洞察は、利用可能な資料が反映するように、2021年と2024/25年の両コンペティションから引き出されます。情報が入手不可能な箇所（例：）については、その限界を認識しています。

C. 中核的課題：制約下でのリアルタイム適応

中心的なテーマは、Kaggle APIの厳格な制約（実行時間制限、時系列モジュールの使用、メモリ使用量）の中で、ほぼリアルタイムに適応できる予測モデルを構築することの難しさです 。これは、後のオンライン学習とパイプラインの堅牢性に関する議論の舞台を設定します。

D. コンペティションの二重の役割：技術的挑戦と人材発掘

いくつかの情報源は、これらのコンペティションがJane Streetにとって洗練された人材発掘イベントおよびマーケティングツールとしても機能している可能性を示唆しており、これが問題設計（匿名化、複雑さ）に影響を与えている可能性があります 。Jane Street自身も、この課題は現実世界の問題を単純化しているものの、同社で行われている作業の種類を紹介するものであると述べています 。コメンテーターは、コンペティションがリクルーティング資料である可能性が高いと指摘しています 。匿名化や特定の制約は、単に知的財産保護のためだけでなく、Jane Streetの業務に関連する特定のスキル（適応性、ノイズの多い/不完全なデータの処理、効率的なコーディング）をテストするために設

計されている可能性があります。この文脈を理解することは、特定の制約（APIなど）が なぜ 存在するのかを理解する助けとなります。参加者は、コンペティションの設定が、実際の取引に関連しつつも、特定のスキルをテストし、潜在的に才能を発見するために設計された人工的な環境でもあることを認識すべきです。これは、単に最高のスタンドアロン取引アルゴリズムを見つけることだけが目的ではないことを示唆しています。

II. モデリングの展望：勾配ブースティングから深層学習の優位性へ

A. 初期戦略：勾配ブースティング決定木（GBDT）

当初、多くの参加者は、表形式データに対する最先端（SOTA）技術としての認識と既存の経験に基づき、GBDT（LightGBM、XGBoost、CatBoost）に依存していました 。しかし、これらのモデルのパフォーマンスには限界がありました。LGBMモデルは、アンサンブルやRidge回帰モデルの追加を行っても、説明分散（R2スコア）で約 0.0045 から 0.0049 の範囲で頭打ちになりました 。XGBoostはLGBMと同等の結果をもたらしましたが、はるかに遅く、CatBoostは一部のテストではLGBMほどの性能を発揮できませんでした 。GBDTとニューラルネットワーク（NN）を比較する議論もあり、GBDTは表形式タスクでしばしば優れているものの、このコンペティションではデータサイズや信号処理の側面からNNが有望視されました 。一部の参加者は、GBDTの可能性がまだ十分に探求されていないと感じていました 。

B. ニューラルネットワーク革命

NN、特に初期には単純な多層パーセプトロン（MLP）を採用すると、パフォーマンスが大幅に向上することが観察されました 。最初のNN提出では、スコアが大幅に上昇しました（例：R2スコアが約 0.005 から 0.0064、あるいは 0.008 へ）。
探求された一般的なNNアーキテクチャは以下の通りです：

- **MLP:** 単純なMLP（例：隠れ層3層、ReLU/Swish活性化関数、ドロップアウト、バッチ正規化）は強力な出発点でした 。比較的小さなネットワーク（例：隠れユニット数 256-128）が、より大きなネットワーク（例：256-128-64）と同様の結果をもたらすこともありました 。
- **オートエンコーダ（AE）：** 次元削減、ノイズ除去、特徴生成に使用され、しばしばMLPと組み合わせて（AE+MLP）用いられました 。2021年の優勝ソリューションは教師ありAE+MLPでした 。
- **リカレントネットワーク（RNN - LSTM/GRU）：** 時系列の側面を探るために検討されましたが、結果はまちまちでした。一部の参加者は改善を得るのに苦労しましたが 、他の参加者はGRUを成功裏に組み込みました 。これはデータ再構築の洞察につながりました 。
- **Transformer:** 同様に探求され、時にはシンボルごとのアテンションが使用されました 。しかし、成功は保証されておらず、一部の参加者は既知のアーキテクチャを試しても当初は肯定的な結果が得られませんでした 。Jamba（Mamba-Transformerハイブリッド）も提案されました 。

C. 比較考察：Jane StreetにおけるGBDT対NN

観察された傾向として、GBDTは強力なベースラインであるものの、NN（特に適切に調整されたMLP、AE+MLP、そして潜在的にシーケンシャルモデル）は、これらの特定のコンペティションにおいてより高いパフォーマンス上限を提供しました 。

進行中の議論として、一般的な研究ではGBDTが表形式データでNNを上回ることが多いとされていますが、データセットのサイズ、不規則性、金融時系列特有の性質（非定常性、信号処理的側面）などの要因が、ここではNNに有利に働いた可能性があります 。オンライン学習の必要性も、勾配更新が高速なNNに有利に働きました 。
NNへの移行は、単なる生のパフォーマンスだけでなく、適応性にも関わっていました。NNは、コンペティションのリアルタイム性とAPI制約によって要求される重要なオンライン学習コンポーネントに対して、より適していることが証明されました。参加者は標準的なGBDTから始めましたが 、パフォーマンスの限界に達しました 。NNへの切り替えは大幅なスコア改善をもたらし 、オンライン学習が不可欠であることが認識されました 。効率的なオンライン学習の実装は、時間制限内でGBDTを再トレーニングするよりもNN（ミニバッチ更新）の方がはるかに容易かつ高速でした 。したがって、この文脈におけるNNの優位性は、複雑なパターンをモデル化する能力と、要求されるオンライン適応メカニズムへの適合性の組み合わせである可能性が高いです。

D. GBDT対NNの比較（Jane Streetコンテキスト）

以下の表は、提供された情報源に基づいて、Jane Streetコンペティションの文脈におけるGBDTとNNの長所と短所をまとめたものです。

特徴	GBDT (LGBM/XGBoost)	ニューラルネットワーク (MLP/AE/RNN/Transformer)
ベースライン性能（表形式）	良好	GBDTに劣る場合あり
パフォーマンス上限（JS）	NNより低い	高い
オンライン学習（JS API）	困難/遅い	適している/速い（ミニバッチ）
トレーニング速度（オフライン）	NNより速い場合あり	GBDTより遅い場合あり
不規則性の処理（一般）	得意	GBDTに劣る可能性
特徴量エンジニアリングとの相互作用	-	AEによるノイズ除去/特徴生成 、補助ターゲット活用
解釈可能性	NNより高い傾向	低い傾向

この表は、コンペティション特有の課題（API、オンライン学習、データ型）の文脈で、議論を簡潔にまとめ、参加者がこの種の問題に関連するトレードオフを迅速に把握するのに役立ちます。

III. 匿名化された世界での特徴量エンジニアリングとデータ前処理

A. 探索的データ分析（EDA）の重要な役割

特徴が匿名化されているにもかかわらず、EDAの重要性はJane Street自身のブログや参加者の経験によって強調されています 。言及されている主なEDAの発見には以下が含まれます:

- 欠損値（NaN）が多い特徴量の特定、特に1日の始まりや約85日目以前 。
- 特徴量の分布、相関、潜在的な時間依存性の観察 。
- ヒストグラムにスパイクがある特徴量の指摘（カテゴリカル特徴量の埋め込みの可能性）。
- レスポンダー（目的変数）の分布と関係性の分析 。
- weight 特徴量の理解（weight=0の取引はスコアリングから除外されるが、トレーニングには有用な可能性がある）。

参照されている特定のEDAノートブック/貢献者には、Carl McBride Ellis 、xhlulu 、Muhammad M. El.Sherbini などがいます。

B. 欠損データ（**NaN**）の処理

一般的な戦略には、前方充填（時系列の性質とNaNのパターンによって正当化される ）、平均/中央値補完 、ゼロ充填 、またはモデルベースの補完（KNNImputerが言及されている ）が含まれます。一部のソリューションでは fillna(0) や過去の日の平均 が使用されました。異なるアプローチが用いられ、時には連続特徴量と潜在的な離散特徴量で区別されました 。2021年の優勝ソリューションでは前方充填が使用されました 。

C. 特徴量の作成と選択

匿名化のため、特徴量エンジニアリングは困難でした 。使用された手法には以下が含まれます：

- 交互作用特徴量**:** 既存の特徴量のペアまたは組み合わせ（例：乗算）を作成し、時にはTabNetの係数などのモデルの説明によって導かれました 。時間制限のため失敗した試みもありました 。
- 集約特徴量**:** 時間ウィンドウにわたる特徴量またはレスポンダーのローリング統計（平均、標準偏差）を計算しました 。既存の特徴量がすでにローリング特徴量であると疑う参加者もいました 。
- ラグ特徴量**:** 特徴量またはレスポンダーの過去の値を使用しました 。
- 対数**/**尖度変換**:** アイデアとして言及されました 。
- 特徴量削除**:** 特徴量（例： で言及された特徴量 9, 10, 11）を削除したり、重要度/EDAに基づいてフィルタリングしたりしました 。異なる分散/トレンドのため、初期の日付（例：85日未満）を削除しました 。
- ノイズ除去**/**表現学習**:** オートエンコーダを使用して、より密で潜在的によりクリーンな特徴表現を作成しました 。AEエンコーダの前にガウスノイズを追加して、データ拡張/正則化を行いました 。

D. モデル用データの構造化

シーケンシャルモデル（NN）向けに、3次元入力 [symbol_id, time_id, feature_id] への移行が記述されています 。データを日ごとのチャンク/テンソルに整理し、潜在的に各日をバッチとして使用することが言及されています 。これはハイエンドGPUで実現可能でした 。

E. 特徴量エンジニアリング：諸刃の剣

一部の参加者はエンジニアリングされた特徴量（交互作用 、AE特徴量 、補助ターゲット ）で成功を収めましたが、他の参加者は最小限の利益しか得られなかったり、実行時制約の問題に直面したりしました 。匿名化は原理に基づいたFEを困難にします 。
この匿名化された文脈での特徴量エンジニアリングは、ハイリスク・ハイリターンです。標準的なFE技術はドメイン知識の欠如により適用が困難です 。一部のチームは交互作用/ラグ/集約特徴量を作成しましたが 、オートエンコーダは暗黙的に特徴量をエンジニアリング/ノイズ除去するために使用されました 。しかし、一部の参加者はFEがあまり役立たなかった、またはAPI制限に対して計算コストが高すぎると指摘しました 。2021年の優勝ソリューションはAE特徴量と補助ターゲットを使用しましたが、それ以外の前処理は比較的単純でした 。
したがって、特に実行時制約を考慮すると、単純で堅牢な前処理（NaN処理、基本的なスケーリング）と強力なモデル（AE+MLPや交互作用を学習できるNNなど）の組み合わせが、複雑な手作り特徴量よりも信頼性の高い道筋である可能性があります。最も影響力のある「特徴量エンジニアリング」は、補助ターゲットの使用であったかもしれません 。

IV. オンライン学習の必要性：モデルのリアルタイム適応

A. オンライン学習の根拠

金融市場は非定常であり 、モデルは変化するダイナミクス、コンセプトドリフト、ボラティリティレジームに適応する必要があります 。コンペティションの構造（予測フェーズ）はこれを明確にテストします 。オンライン学習は、このような環境でモデルの予測性能を維持・向上させるために不可欠と見なされました。

B. 実装戦略とフレームワーク

- 頻度: APIでラベルが受信される方法に合わせて、日次更新が一般的でした 。
- データウィンドウ: 更新には最近のデータ（例：過去15日間 、20-50日間 ）を使用し、時には古いデータの小さな割合と混合しました 。スライディングウィンドウが使用されました 。
- モデルタイプ: GBDTの再トレーニングと比較して、ミニバッチ/勾配ステップによる更新が高速なため、NNがより適していました 。GBDTのオンライン学習（LGBM refit() または数日ごとの再トレーニング）も試みられましたが、効果が低いか、多様化に寄与しないことがわかりました 。
- トレーニング: オンラインフェーズでは、しばしばエポック数を減らし（例：毎日2エポック ）、学習率を小さくしました 。
- 実装: データキャッシング、状態管理（ステートフルモデルの場合）、時系列APIとの統合を処理するために、慎重なパイプライン設計が必要でした 。バッチ処理には tf.keras.utils.Sequence のようなツールの使用が提案されました 。

C. 直面した重大な課題

- 実行時制約: 予測ループごとの厳格な時間制限（例：2024/25年で言及された約1分 、合計5/8時間制限 ）により、広範な再トレーニングは実行不可能でした 。勾配更新がしばしばボトルネックとなりました 。
- 実装の複雑さ: パイプライン（データ読み込み、状態管理、APIインタラクション）を正しく構築することは簡単ではなく、バグが発生しやすい状況でした 。デバッグは困難でした 。
- パフォーマンス低下: オンライン学習がスコアを悪化させることがありました（「壊滅的な忘却」、負のR2）。更新パラメータ（LR、エポック、バッチサイズ）の調整は重要ですが困難でした 。
- フレームワークの問題: 互換性の問題（例：Keras 3のRagged Tensorサポート ）、カスタム実装の困難さ 、機能不足（例：LGBMトレーニングのタイムアウトパラメータ ）がありました。柔軟性のためにPyTorchがしばしば好まれました 。
- **LGBM**特有の問題: タイムアウトパラメータの欠如、部分トレーニングの起動時間の線形増加 。効果的な再トレーニングの難しさ 。

D. 成功の証拠（機能した場合）

オンライン学習が成功裏に実装された場合、控えめながらも肯定的なスコア改善が報告されました（例：NNによる0.0006 R2の向上 、LGBMの多様化によるスコア改善 、0.0043から0.0058 R2へのジャンプ ）。2024/25年の10位チームは、成功したオンラインNN実装について明確に議論しました 。12位チームも日次更新を実装していました 。

E. オンライン学習：必要悪か競争優位性か？

問題の性質上、オンライン学習は普遍的に重要と見なされていましたが、その成功裏の実装は大きな差別化要因であり、重大なフラストレーションの原因でもありました。多くの参加者が制約内で効果的に機能させるのに苦労しました。
問題は非定常性のため適応を要求し 、コンペティション形式（予測フェーズ）はこれを直接テストします 。参加者は早い段階でその重要性を認識していましたが 、実装には深刻な技術的ハードル（API制限、コーディングの複雑さ、フレームワークの問題、パフォーマンス低下のリスク）がありました 。成功したオンライン学習はスコア向上をもたらし、トップチームにとって重要な要因であったことを示唆しています 。多くの参加者はうまく機能させることができず、パフォーマンスが妨げられました 。したがって、コンペティションの制約内でオンライン学習を習得することは、間違いなく最大の技術的課題の1つであり、成功の主要な決定要因でした。これには、優れたモデリングのアイデアだけでなく、強力なエンジニアリングスキル（パイプライン設計、最適化、デバッグ）と慎重な調整が必要でした。これは単なる機能ではなく、トップレベルで競争するために必要なコアコンピテンシーでした。

F. オンライン学習の課題と報告された解決策/緩和策

以下の表は、オンライン学習における主な課題と、参加者がそれらにどのように対処しようとしたかをまとめたものです。

課題	説明（情報源より）	報告された解決策/アプローチ（情報源より）	関連情報源
実行時間制限	ステップあたり1分、合計8時間など	NNミニバッチ、日次更新、パイプライン最適化、慎重なアーキテクチャ選択	
実装の複雑さ	パイプライン構築、状態管理、API連携が困難、バグが多い	オフラインでのAPI模倣、段階的なデバッグ、堅牢なパイプライン設計、PyTorchなどの柔軟なフレームワーク利用	
パフォーマンス低下（壊滅的忘却）	スコア悪化、学習済み重みの喪失	慎重なLR/エポック調整、オフラインでのAPI模倣、日次更新、補助ターゲットの使用？	
フレームワークの問題	Keras互換性問題、カスタム実装の困難さ、LGBM機能不足	PyTorchへの移行、適切なバージョン管理、問題に合わせたフレームワーク選択	
GBDTの非現実性	時間制限内での再トレーニングが困難	NNへの移行	

この表は、オンライン学習が重要でありながら困難な側面であることを示しています。参加者が直面した具体的な問題を統合し、対応する試みられた解決策を提供することで、将来の参加者にとって実用的なガイドとなります。

V. 堅牢性の確保：検証、過学習、リーダーボードの試練

A. 時系列クロスバリデーション（CV）戦略

時間依存性のため、特殊なCVが必要であることが強調されました。標準的なKFoldは不適切です 。一般的なアプローチには以下が含まれます：

- グループ化時系列分割: 時間ベースのグループ（例：日）を使用して分割し、テストフォールドが常にトレインフォールドより時間的に後になるようにします 。
- パージ分割/ギャップ: ラグ特徴量や移動平均からのリークを防ぐために、トレインセットとバリデーションセットの間にギャップを導入します 。10日間のギャップが言及されました 。
- ウォークフォワード検証: 拡大またはスライディングウィンドウを使用し、実際の展開を模倣します（ はウォークフォワードに言及）。最後のN日間（例：100日間 、120日間 ）を固定バリデーションセットとして使用します。

B. 過学習の蔓延する脅威

低いシグナル対ノイズ比と非定常性により、過学習が大きな懸念事項であることが繰り返し述べられました 。基本的な検証以外に使用された手法には以下が含まれます：

- 正則化: NNで一般的に使用されるドロップアウト、バッチ正規化 。入力へのガウスノイズの追加 。ユーティリティスコアに基づくファインチューニング正則化 。
- 単純なモデル: より単純なモデルの方がパフォーマンスが良い、または過学習しにくいと考える参加者もいました 。
- アンサンブル: 異なるシードや異なるデータ分割/フォールドでトレーニングされたモデルを平均化して分散を減らしました 。
- 慎重な特徴量エンジニアリング: 過度に複雑または潜在的にリークのある特徴量を避けました。AEによるノイズ除去 。
- 堅牢性への焦点: パブリックLBスコアの最大化よりも堅牢性を明示的に優先する参加者もいました 。

C. リーダーボードのダイナミクスと変動

パブリックリーダーボードとプライベートリーダーボードのスコアの乖離（「シェイクアップ」）現象について議論され、パブリックLBへの過学習が一般的であることが示唆されました 。2021年のコンペティションでは評価期間が長く、スコアの変動が見られました 。2021年のコンペティション中、プライベートLBのフィードバックが限られていたため、真のパフォーマンスを知ることが困難であったことが言及されました 。2024/25年の予測フェーズでは、一部のチームでスコアリングに影響を与える技術的な問題（再実行の失敗、提出成功にもかかわらずスコアリングされない）が報告されました 。

D. 検証は必要だが十分ではない

適切な時系列CVを使用しているにもかかわらず、過学習とリーダーボードの変動は依然として重大な問題でした。成功には、堅牢性に対する多面的なアプローチが必要でした。参加者はギャップ付きの時系列CVを実装しましたが 、過学習は依然として主要なリスクであり、プライベートLBのパフォーマンス低下の可能性のある原因として挙げられました 。勝利したソリューションでは、アンサンブル、正則化、ノイズ注入、時にはより単純なアーキテクチャなどの追加技術がしばしば採用されました 。時間の経過に伴うスコアの変動 やパブリック/プライベートの不一致 は、良好なCVスコアが最終的な成功を保証しなかったことを裏付けています。
この領域での堅牢性は、単なる優れた検証スキーム以上を要求します。固有のノイズと非定常性に対抗するためには、適切なCV、モデル正則化、アンサンブル、潜在的により単純なアーキテクチャ、

そしておそらくドメイン固有の技術（特定のターゲットに焦点を当てる、AEノイズ除去を使用するなど）の組み合わせが必要です。パブリックLBスコアを信頼することは危険です。

VI. ケーススタディ：勝利ソリューションの解剖（2021年1位 AE+MLP）

A. アプローチの概要

2021年コンペティションの1位ソリューション（Yirun Zhang氏）は、教師ありオートエンコーダ（Supervised Autoencoder）とMLPを組み合わせ、最終提出ではチームメイトのXGBoostモデルとブレンドされました 。単一のAE-MLPモデルだけでも、プライベートリーダーボードの再実行で1位を確保しました 。このアーキテクチャは他の参加者によっても採用され 、Numeraiのような類似の問題との関連で議論されました 。

B. アーキテクチャの詳細

- 教師ありオートエンコーダ: その目的は、予測ターゲットによって導かれながら新しい特徴量を作成（ノイズ除去/次元削減）することでした 。ターゲット情報は勾配のショートカットとして機能し、より関連性の高い特徴生成を強制します 。デコーダ出力もターゲット予測に使用されました 。
- **MLP:** 下流のMLPは、元の特徴量とAEのエンコードされた特徴量を連結したものを入力として受け取ります 。
- 共同トレーニング: AEがバリデーションセットと重複するデータで事前トレーニングされた場合に発生する可能性のあるラベルリークを防ぐため、AEとMLPは各*CV*フォールド内で一緒にトレーニングされました 。
- 正則化/拡張: エンコーダの前にガウスノイズ層が追加されました 。MLPではバッチ正規化とドロップアウトが使用されました 。ReLUよりもSwish活性化関数が好まれました 。

C. 特徴量の取り扱いと前処理

- トレーニングのために最初の85日間を削除しました 。
- 欠損値には前方充填を使用しました 。
- すべてのrespターゲット（resp, resp_1からresp_4）を多ラベル分類のためにバイナリアクション（action > 0）に変換しました 。
- トレーニング中のサンプル重みとして、すべてのrespターゲットの絶対値の平均を使用し、影響の大きいサンプルにモデルが焦点を当てるようにしました 。
- 推論時：すべてのターゲット出力から予測されたアクションの平均を取りました 。

D. トレーニングと検証戦略

- 31日間のギャップを持つ5分割パージグループ時系列分割を使用しました 。
- 最終アンサンブルには、最後の2つのCV分割でトレーニングされたモデル（より多くのデータでトレーニングされた）のみを使用しました 。
- 3つの異なるランダムシードでモデルをトレーニングし、予測を平均化しました 。
- ハイパーパラメータ調整にはHyperoptを使用しました 。
- 早期停止のためには、全体の損失ではなくMLPのBCE損失のみを監視しました 。

E. 移転可能な教訓

- 教師あり**AE:** ノイズが多く高次元なデータにおいて、ノイズ除去、特徴抽出、予測を統合するための強力な手法です。
- 補助ターゲット**:** 主要ターゲット（resp）とともに関連する複数のターゲット（resp_1からresp_4）を使用することでパフォーマンスが向上しました（他の参加者も指摘 ）。
- 慎重な**CV:** AEの事前トレーニングや特徴生成のような手法を使用する場合、リークを防ぐためにはパージ/ギャップを伴う厳密な時系列CVが不可欠です。フォールド内でAEをトレーニングすることが重要です。
- 正則化**:** 複数の手法（ノイズ、ドロップアウト、バッチ正規化）を組み合わせることが鍵です。
- アンサンブル**:** シード間、そして潜在的にフォールド間で平均化することで堅牢性が向上します。

F. 統合アーキテクチャの力

勝利したソリューションは単なるMLPではありませんでした。それは、AEコンポーネントが教師あり信号によって導かれ、よりクリーンで関連性の高い特徴を提供することでMLPを積極的に支援する統合システムでした。
問題はノイズが多く、高次元で、匿名化された特徴量を含んでいました 。単純なMLPは合理的に機能しましたが、改善の余地がありました 。オートエンコーダはノイズ除去と表現学習で知られています 。勝利したソリューションはAEを教師ありで学習させ、単なる再構築ではなく予測タスクに役立つ表現を学習するように強制しました 。この統合（AEがMLPに入力を提供し、共同でトレーニングされ、教師ありAE損失を使用）は相乗効果を生み出し、より単純なモデルを上回りました。この技術は、ノイズ、高次元性、特徴量の関連性という複数の課題に同時に取り組みます。
Jane Streetのような複雑でノイズの多いデータセットに対しては、表現学習（AEなど）を教師あり予測タスクと直接統合し、それらを共同でトレーニングするアーキテクチャが、シーケンシャルなアプローチ（例：最初にAEをトレーニングし、次にMLPをトレーニングする）よりも大幅な利点を提供する可能性があります。AEの教師あり側面が特に重要であるように思われます。

VII. 結論：高頻度取引予測に取り組むための主要な学び

A. 主要戦略の統合

GBDTに対するNN（MLP、AE+MLP、潜在的にシーケンシャルモデル）の優位性を要約します。これは、パフォーマンス上限とオンライン学習への適合性の両方によって推進されました。API制約内でのオンライン学習/適応の重要でありながら困難な役割を再確認します。過学習に対抗するための堅牢な時系列検証、多面的な正則化、アンサンブルの重要性を要約します。特徴量向上のための補助ターゲットと教師ありAEのような統合アーキテクチャの価値を強調します。

B. エンジニアリングと堅牢性の重視

これらのコンペティションでの成功は、強力なMLモデリングスキルだけでなく、厳格なAPI制約に従う信頼性が高く効率的なパイプラインを構築するための堅牢なソフトウェアエンジニアリングの実践も必要とすることを強調します 。特にノイズの多いデータと長いフィードバックループを考慮すると、忍耐力と系統的な実験の必要性を強調します 。

C. 最終的な考察

問題固有の困難さと低いシグナル対ノイズ比を認識します 。単一の「特効薬」は存在しないものの、適切なNNアーキテクチャ、慎重な特徴量処理（AE/補助ターゲットを含む）、堅牢な検証、効果的なオンライン適応、そして強力なエンジニアリングの組み合わせが、提供された情報源に記録された集合的な経験に基づいて、成功への最も有望な道筋を提供すると結論付けます。

引用文献

1. Jane Street Market Prediction | Kaggle, https://www.kaggle.com/competitions/jane-street-market-prediction 2. Jane Street Real-Time Market Data Forecasting | Kaggle, https://www.kaggle.com/competitions/jane-street-real-time-market-data-forecasting/discussion/556548 3. [N] Kaggle/Jane Street Market Prediction Challenge : r/MachineLearning - Reddit, https://www.reddit.com/r/MachineLearning/comments/jzndmq/n_kagglejane_street_market_prediction_challenge/ 4. Jane Street Market Prediction ($100k Kaggle competition) - Hacker News, https://news.ycombinator.com/item?id=25201326 5. codefluence/jane_street: kaggle competition - GitHub, https://github.com/codefluence/jane_street 6. Jane Street Market EDA | Jaekang AI, https://leejaeka.github.io/jaekangai/python/eda/jane%20street/kaggle/visualization/big%20data/2021/01/23/JaneStreet-Copy1.html 7. abdelghanibelgaid/Jane-Street-Market-Prediction: MLP ... - GitHub, https://github.com/abdelghanibelgaid/Jane-Street-Market-Prediction 8. scaomath/kaggle-jane-street: Machine learning models to predict realtime financial market data provided by Jane Street - GitHub, https://github.com/scaomath/kaggle-jane-street 9. Jane Street: t-SNE using RAPIDS cuML - Kaggle, https://www.kaggle.com/code/carlmcbrideellis/jane-street-t-sne-using-rapids-cuml 10. Autoencoder Explained - YouTube, https://www.youtube.com/watch?v=H1AllrJ-_30 11. Yirun's Solution (1st place): Training Supervised Autoencoder with MLP - Jane Street Market Prediction | Kaggle, https://www.kaggle.com/c/jane-street-market-prediction/discussion/224348 12. Online learning - Jane Street Real-Time Market Data Forecasting | Kaggle, https://www.kaggle.com/competitions/jane-street-real-time-market-data-forecasting/discussion/549746 13. Jane Street Market Prediction | Kaggle : r/quant - Reddit, https://www.reddit.com/r/quant/comments/jzoayx/jane_street_market_prediction_kaggle/ 14. A Data-Centric Perspective on Evaluating Machine Learning Models for Tabular Data - OpenReview, https://openreview.net/pdf?id=kWTvdSSH5W 15. When Do Neural Nets Outperform Boosted Trees on Tabular Data? - arXiv, https://arxiv.org/pdf/2305.02997 16. [D] Can we begin to understand possible mathematical reasons as to why algorithms like "xgboost" and "random forest" win Kaggle Competitions, instead of neural networks? - Reddit, https://www.reddit.com/r/MachineLearning/comments/nxn65x/d_can_we_begin_to_understand_possible/ 17. AutoEncoder and multitask MLP on new dataset (from Kaggle Jane Street) - Numerai Forum, https://forum.numer.ai/t/autoencoder-and-multitask-mlp-on-new-dataset-from-kaggle-jane-street/4338 18. Jane Street Market Prediction | Jaekang AI, https://leejaeka.github.io/jaekangai/mlp/python/feature%20engineering/imputation/jane%20street/kaggle/visualization/big%20data/random%20forest/2021/01/25/jane-predict.html 19. Jane Street 1st place solution -> Crypto PyTorch - Kaggle, https://www.kaggle.com/code/neodory/jane-street-1st-place-solution-crypto-pytorch 20.

Semi-Supervised Anomaly Detection Explained with Autoencoders - YouTube, https://www.youtube.com/watch?v=IY4M-cnPMOU 21. Supervised Autoencoder MLP for Financial Time Series Forecasting - arXiv, https://arxiv.org/html/2404.01866v1 22. Supervised autoencoders: Improving generalization performance with unsupervised regularizers - NIPS papers, https://proceedings.neurips.cc/paper/7296-supervised-autoencoders-improving-generalization-performance-with-unsupervised-regularizers.pdf 23. Jane Street Keras Notebook - LSTM, https://www.kaggle.com/code/nicholashojx/jane-street-keras-notebook-lstm 24. [Proposal-ML] ***** | OpenReview, https://openreview.net/forum?id=1tVpbWoJRX 25. Jane Street: EDA of day 0 and feature importance - Kaggle, https://www.kaggle.com/code/carlmcbrideellis/jane-street-eda-of-day-0-and-feature-importance 26. Jane Street'24 : Day 0 EDA and feature importance - Kaggle, https://www.kaggle.com/code/ahsuna123/jane-street-24-day-0-eda-and-feature-importance 27. Jane Street Keras Notebook - MLP - Kaggle, https://www.kaggle.com/code/nicholashojx/jane-street-keras-notebook-mlp 28. Collection of EDA - Kaggle, https://www.kaggle.com/code/saurabhshahane/collection-of-eda 29. Carl McBride Ellis | Discussion Grandmaster - Kaggle, https://www.kaggle.com/carlmcbrideellis/discussion 30. Jane Street: Time series analysis + EDA + Ensemble - Kaggle, https://www.kaggle.com/code/mostafa3apre7em/jane-street-time-series-analysis-eda-ensemble/notebook 31. Victor Shlepov | Discussion Expert - Kaggle, https://www.kaggle.com/victorshlepov/discussion