

電柱の保守予測を最適化して サービス品質を向上

インテル® AI アナリティクス・ツールキットでマシンラーニングを
高速化

Kelli Belcher インテル コーポレーション AI ソリューション・エンジニア

世界的に電力需要が増加する中、信頼性の高いサービスを提供し、コストのかかる予期しないダウンタイムを回避するには、インフラ設備を正確かつ効率的に監視する必要があります。現在の手動による問題の特定は50%未達の精度であり、電柱の保守と交換のコストは100億米ドルを超えています（出典：[Utility Poles : Maintenance or Replacement](#)（英語）、Utility Partners of America、2020年8月8日）。

インテルはAccentureと協力して、[Predictive Asset Analytics Reference Kit](#)（英語）を開発しました。これは、インフラ設備を予防的に保守し、システムの安定性を向上し、機能停止やダウンタイムを回避し、運用コストを抑えるため、インフラ設備の健全性と障害の可能性を予測するように設計されています。このリファレンス・キットには、トレーニング・データ、オープンソースのトレーニング済みの資産予測分析モデル、ライブラリー、ユーザーガイド、およびトレーニング・サイクル、予測スループット、精度を最適化するoneAPIコンポーネントが含まれています。

このリファレンス・キットの実装方法を示すため、このチュートリアルで使用するデータセットは、[ユーザーガイド](#)（英語）の手順に従って生成されました。ワークフローを**図 1** に示します。これは、電柱の使用年数、保守履歴、機能停止記録、地理空間データなど、インフラ設備の全体的な健全性に関する 34 の特徴で構成されています。ターゲット変数は、電柱が故障しているかどうかを示すバイナリー・インジケータです。

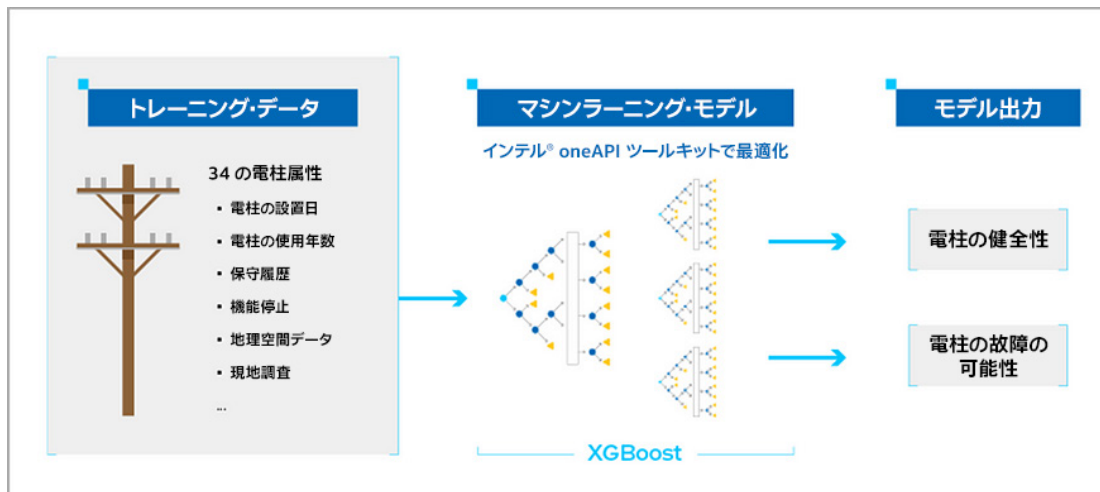


図 1. Predictive Asset Analytics Reference Kit のワークフロー

このリファレンス・キットの主な利点は、コードをほとんどまたは全く変更することなく、ヘテロジニアスな XPU アーキテクチャー全体でモデルのトレーニングと推論を最適化できることです。[インテル® AI アナリティクス・ツールキット \(AI キット\)](#) と [インテル® oneAPI データ・アナリティクス・ライブラリー \(インテル® oneDAL\)](#) がこれを可能にします。このガイドで使用する AI キットの主なライブラリーは、[Modin 向けインテル® ディストリビューション](#)（英語）、[scikit-learn* 向けインテル® エクステンション](#)（英語）、[XGBoost 向けインテル® オプティマイゼーション](#)（英語）です。これらはすべて、AI キットの一部またはスタンドアロン・ライブラリーとしてダウンロードできます。さらに、インテル® oneDAL の [daal4py](#)（英語）を使用して、XGBoost モデルの推論を高速化します。

データ処理

まず、Modin を使用してデータを処理および調査します。Modin は、pandas ワークフローをデータセットのサイズに合わせてスケーリングするように設計された分散型 DataFrame ライブラリーであり、1MB ~ 1TB 以上のデータセットをサポートします。pandas では、一度に 1 つのコアのみが使用されます。Modin の Dask エンジンは、利用可能なすべての CPU コアを利用できるため、非常に大きなデータセットをはるかに高速に処理できます。

図 2 は、2GB のデータセットに対して同じ pandas 操作を実行する Modin (左) と pandas (右) の例です。2 つの例の唯一の違いは、インポート文です。

	
<pre>In [2]: import modin.pandas as pd In [3]: %time df = pd.read_csv("taxi.csv", parse_dates=["tpep_pickup_datetime", "tpep_dropoff_datetime"], quoting=3) CPU times: user 1.57 s, sys: 683 ms, total: 2.26 s Wall time: 14.2 s In [4]: %time isnull = df.isnull() CPU times: user 138 ms, sys: 27.3 ms, total: 166 ms Wall time: 404 ms In [5]: %time rounded_trip_distance = df[["pickup_longitude"]].applymap(round) CPU times: user 175 ms, sys: 28.4 ms, total: 203 ms Wall time: 663 ms In []:</pre>	<pre>In [2]: import pandas as pd In [3]: %time df = pd.read_csv("taxi.csv", parse_dates=["tpep_pickup_datetime", "tpep_dropoff_datetime"], quoting=3) CPU times: user 30.7 s, sys: 4.25 s, total: 35 s Wall time: 35.3 s In [4]: %time isnull = df.isnull() CPU times: user 1.37 s, sys: 300 ms, total: 1.67 s Wall time: 1.67 s In [5]: %time rounded_trip_distance = df[["pickup_longitude"]].applymap(round) CPU times: user 3.07 s, sys: 305 ms, total: 3.37 s Wall time: 3.37 s In []:</pre>

図 2. Modin (左) と pandas (右) のパフォーマンスの比較

Dask エンジンで Modin を使用するには、次のコードセルに示すように、最初に pandas のドロップイン置換をインポートしてから、エンジン呼び出し文で Dask 実行環境を初期化します。

```
import modin.pandas as pd
from modin.config import Engine
Engine.put("dask")
```

探索的データ分析

Modin の Dask エンジンが初期化されると、利用可能な CPU コア間で並列に処理される同じ pandas 関数を使用して、データの処理と調査を続行できます。図 3 の State 1 に示すように、データ内の電柱の約 40% に不具合があると識別されました。ターゲット分布のわずかな不均衡を考慮して、交差検証中に層化抽出法を使用します。データをさらに詳しく調べるには、[こちらのノートブック](#) (英語) を参照してください。

ターゲット分布

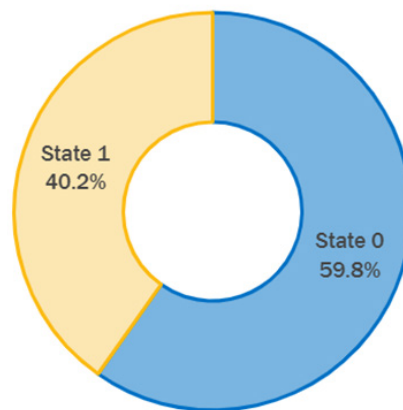


図 3. ターゲット変数 Asset Label の分布

ソリューションの構築

電柱の故障の確率を予測するため、いくつかのマシンラーニング・アルゴリズムを検討しました。データの非線形性のため、サポートベクトル分類器 (SVC) と XGBoost の 2 つのノンパラメトリック・モデルを選択しました。どちらも、特徴とターゲット変数の間の非線形関係を捉えることができる高性能の分類器です。しかし、ハイパーパラメーターをチューニングし、大規模な産業用データセットのモデルをトレーニングするには、長い時間がかかります。ここで、インテル® AI アナリティクス・ツールキットの出番です。

サポートベクトル分類器による分析

SVC モデルのトレーニングと推論を最適化するため、scikit-learn* 向けインテル® エクステンションを利用して、インテルの CPU と GPU の両方で、ベクトル命令、スレッド化、メモリー最適化を使用することで、アルゴリズムの実行時間を短縮します。scikit-learn* 向けインテル® エクステンションは、サポートベクトルマシン、K 近傍法、ランダムフォレストなど、scikit-learn* ライブラリー内の多くの推定器と関数をサポートしています。

これを実装するには、いくつかの方法があります。サポートされているすべてのアルゴリズムをグローバルに動的にパッチするには、以下のコードセルに示す関数を呼び出すだけです。

```
from sklearnex import patch_sklearn
patch_sklearn()
```

このパッチは、サポートされている stock scikit-learn* アルゴリズムを最適化されたバージョンに置き換えます。そして、コードにほかの変更を加えることなく、同じ AI パッケージと scikit-learn* ライブラリーを引き続きインポートして使用できます。scikit-learn* 向けインテル® エクステンションの適用に関する詳細は、『[デベロッパーガイド](#)』（英語）を参照してください。

層化 3 分割交差検証を使用して SVC をチューニングし、見つかった最良のハイパーパラメーターは、正則化パラメーターが 10 の放射基底関数カーネルでした。サンプル外のテストセットでは、SVC は 0.919 の ROC 曲線下面積 (AUC) と 0.902 の平均精度 (AUPRC) スコアを達成しました (図 4)。

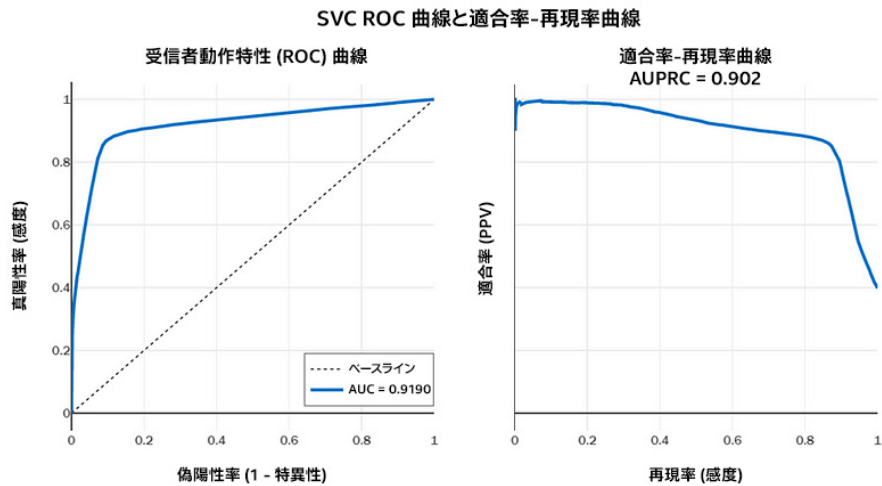


図 4. テストケースの SVC パフォーマンス

XGBoost 向け Intel® オプティマイゼーションによる分析

SVC のパフォーマンスをさらに向上できるかどうか確認するため、層化 3 分割交差検証を使用して、上記と同じ手順で XGBoost モデルをチューニングしました。リファレンス・キットの開発には、XGBoost v1.4.3 が使用されました。XGBoost v0.81 以降には、Intel がパッケージに直接アップストリームしている最適化が含まれています。次のグラフは、このデータセットのデフォルトのハイパーパラメータを使用した stock XGBoost v0.81 と比較した、XGBoost v1.4.3 のトレーニング (図 5) と予測 (図 6) 時間の向上を示しています。

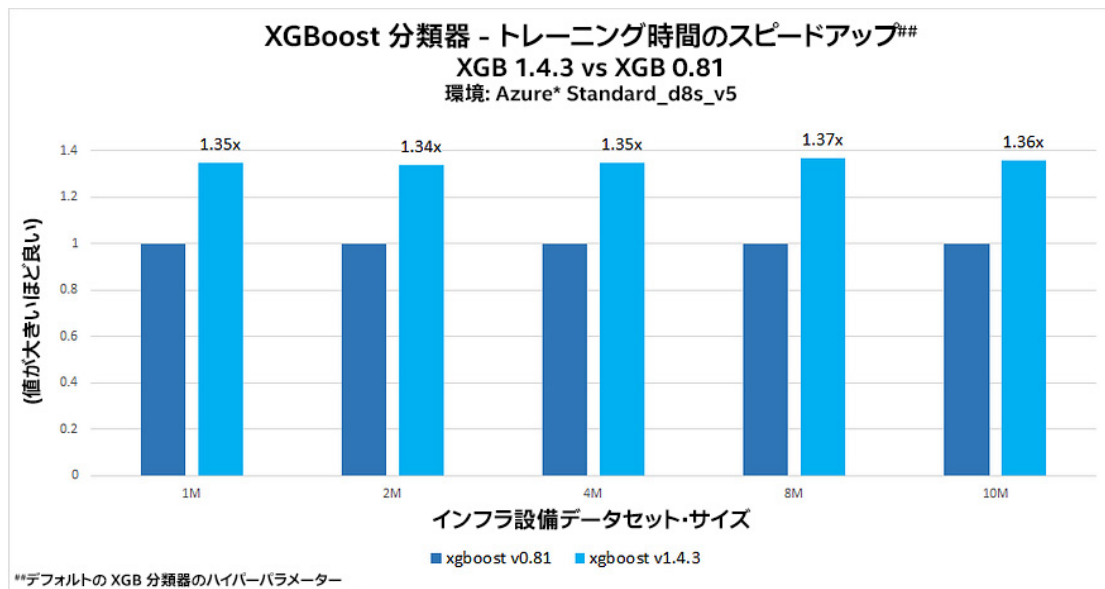


図 5. XGBoost v1.4.3 で利用可能な Intel の最適化は stock XGBoost v0.81 と比較してトレーニングを約 1.37 倍高速化

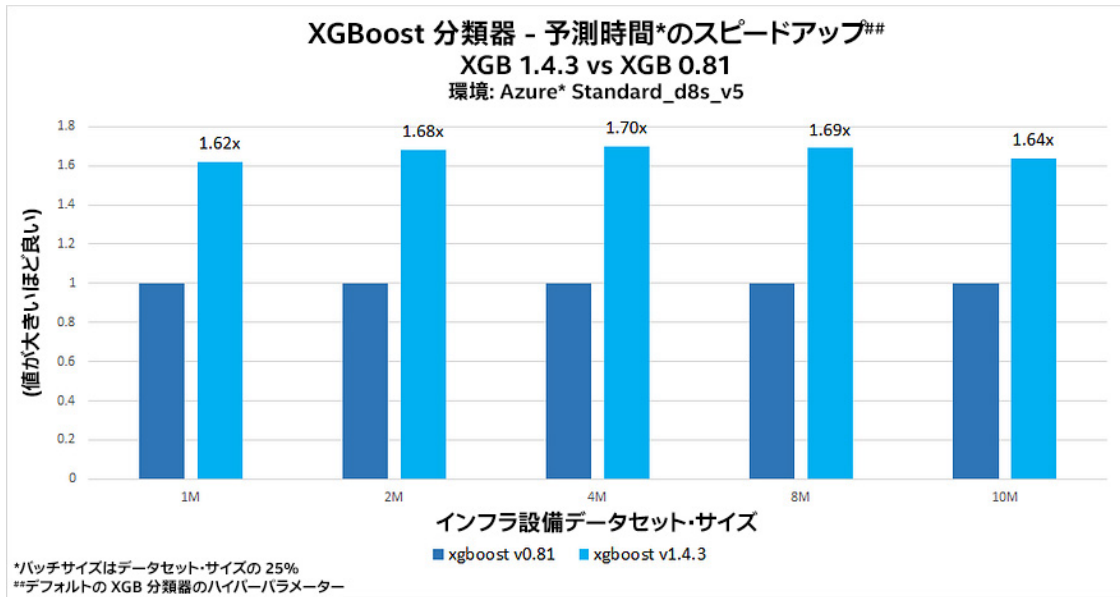


図 6. XGBoost v1.4.3 で利用可能なインテルの最適化は stock XGBoost v0.81 と比較して予測時間を 1.62 倍～ 1.70 倍向上

予測時間をさらに向上するため、チューニングした XGBoost モデルを daal4py モデルに変換できます。daal4py は、[oneDAL](#)（英語）向けの Python* API です。インテル® AVX-512 のベクトル化を使用して、インテル® Xeon® プロセッサで勾配ブースティングのパフォーマンスを最大化します（図 7）。

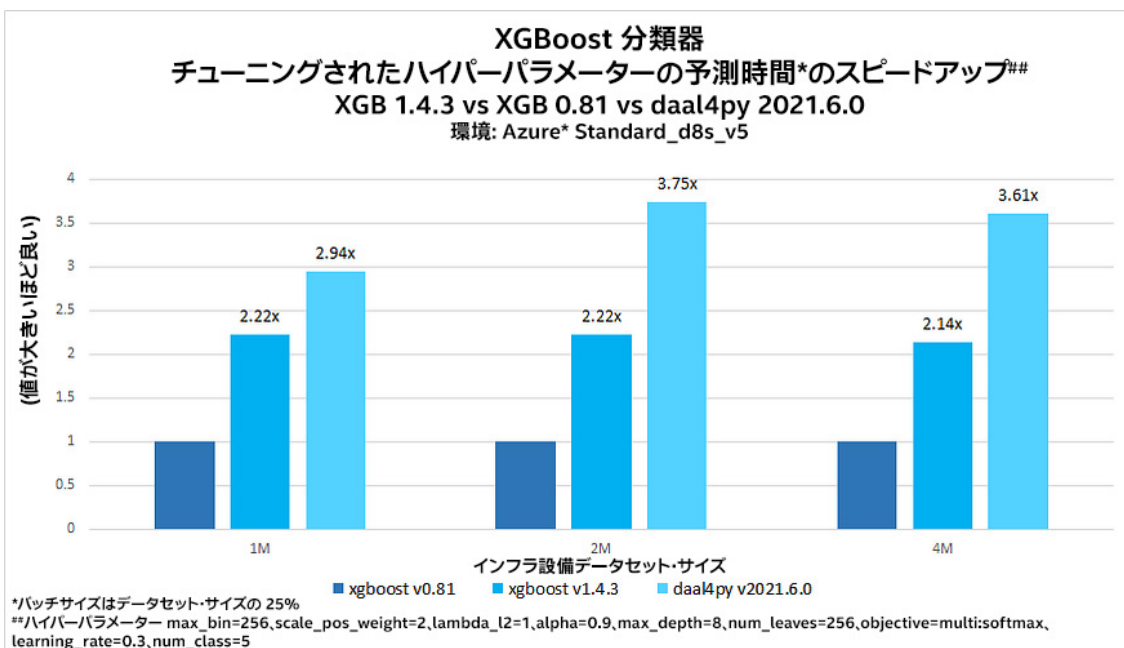


図 7. stock XGBoost v0.81 と比較した XGBoost v1.4.3 と daal4py v2021.6.0 で利用可能なインテルの最適化による予測時間のスピードアップ

チューニングしたハイパーパラメーターにより、XGBoost 向けインテル® オプティマイゼーションは、2.14 倍~ 2.22 倍のスピードアップを実現し、予測時間を短縮しました。daal4py は予測時間をさらに向上し、XGBoost v0.81 と比較して 2.94 倍~ 3.75 倍の高速化をもたらしました。チューニングした XGBoost モデルから daal4py への変換は、以下のコード行で簡単に行えます。

```
import daal4py as d4p
daal_model = d4p.get_gbt_model_from_xgboost(xgb.get_booster())
```

そして、トレーニング済みモデルを入力データとともに daal4py の予測関数に送り、テストセットでの確率を計算できます。

```
daal_prob = d4p.gbt_classification_prediction(nClasses = 2, resultsToEvaluate = "computeClassLabels|computeClassProbabilities",
fptype = 'float').compute(X_test, daal_model).probabilities
```

図 8 は、サンプル外のテストセットにおいて、daal4py により高速化した XGBoost モデルが達成した予測パフォーマンスを示しています。XGBoost は、AUC が 0.9401、平均精度が 0.937 と、SVC を凌駕する結果を示しました。

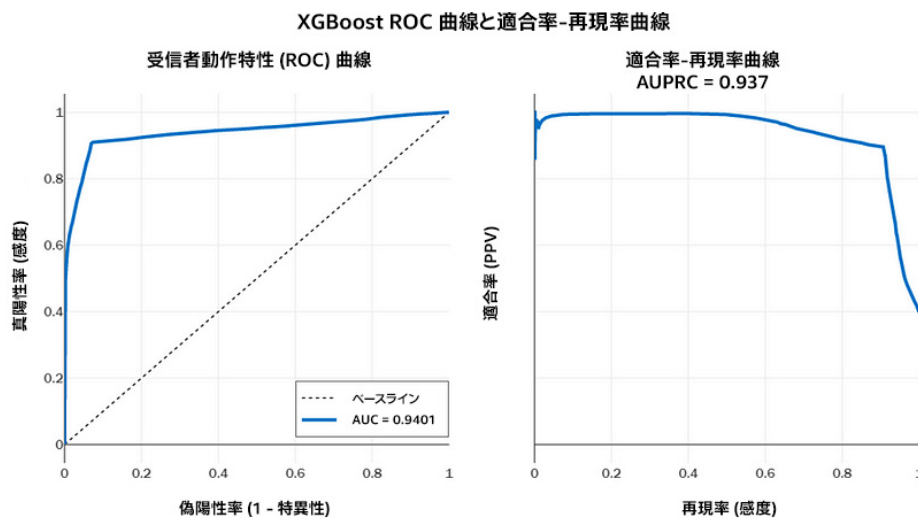


図 8. daal4py 最適化を使用したテストセットでの XGBoost のパフォーマンス

まとめ

大規模な保守予測ソリューションでは、通常、トレーニングと推論の高速化が必要です。インテル® AI アナリティクス・ツールキットで利用可能な最適化を使用して、電柱の故障の可能性を予測する効率良いエンドツーエンドの確率的分類ツールを開発しました。トレーニングと推論を高速化した結果、数十万件のインフラ設備の健全性予測を作成する計算時間の短縮、生産性の向上、コスト軽減を実現しました。このリファレンス・キットは、同様の業界やユースケースに簡単にスケーリングできる、公益事業者向けの設備保守予測のパフォーマンス最適化ガイドを提供します。