

# ⑤ 必須：MLモデル 成果物レポート（Sample）

## 1. 基本情報

- 受講者氏名：田中 太郎
- 作成日：2025年10月1日
- 対象工程：⑤ 必須（MLモデル構築）
- 成果物ファイル名：power\_demand\_prediction\_models.py
- 成果物バージョン：1.0
- 対応環境（OS / 言語 / ライブラリ / バージョン）：
  - OS: Windows 11, macOS 12+, Ubuntu 20.04+
  - 言語: Python 3.10.11
  - 主要ライブラリ: TensorFlow 2.13+, LightGBM 4.0+, PyCaret 3.2+, scikit-learn 1.3+
  - 依存関係: pandas 2.0+, numpy 1.24+, matplotlib 3.7+, psutil 5.9+

## 2. 成果物概要

- 成果物の目的：電力需要の高精度予測を通じた電力安定供給とエネルギー効率最適化を実現するAIモデル群の構築
- 主な機能：
  - 4種類のMLアルゴリズム（Keras, LightGBM, PyCaret, RandomForest）による包括的予測システム
  - 時系列特徴量エンジニアリング（月・週・時間・気温による多次元予測）
  - 自動データ取得・前処理・モデル学習・予測実行の統合ワークフロー
- 対象データや入力条件：
  - 東京電力管内の電力需要実績データ（2016年～2025年）
  - 気温データ（Open-Meteo API経由）
  - 入力特徴量: MONTH(1-12), WEEK(0-6), HOUR(0-23), TEMP(摂氏温度)
  - 目標変数: KW(電力需要量)
- 出力内容：
  - 翌日電力需要予測値（4モデル独立予測）
  - 予測精度評価指標（RMSE, R2スコア）
  - 可視化レポート（時系列グラフ、散布図、学習曲線）
  - パフォーマンス監視ログ（実行時間、メモリ使用量）
- 想定利用者：電力オペレーター、エネルギー分析者、システム管理者、開発エンジニア

## 3. 作成手順概要

- 元となる仕様書・要件定義の確認内容：
  - 電力需要予測AIモデル構築要件定義書（668行）
  - 予測精度目標（RMSE < 200kW, R2スコア > 0.9）
  - 処理性能目標（メモリ使用量50%削減、学習時間30%短縮）
- 実装時に行った追加調査や分析：
  - TEPCO電力使用実績データの構造解析
  - Open-Meteo API仕様調査とデータ取得方法検証
  - 4種類のMLアルゴリズムの特性比較分析
  - float32メモリ最適化効果の定量評価
- 使用した生成AIモデル（例：GPT-4.1, GPT-4o 等）：
  - 設計フェーズ: GPT-4o (アーキテクチャ設計、モジュール分離戦略)
  - 実装フェーズ: GPT-4.1 (コード生成、デバッグ支援)
  - 最適化フェーズ: Claude-4-Sonnet (パフォーマンス改善提案)
- 作成時に参照した資料や仕様書：
  - TensorFlow/Keras公式ドキュメント
  - LightGBM Performance Tuning Guide
  - PyCaret Auto ML Best Practices
  - scikit-learn Time Series Analysis Guide

## 4. 要件適合性

- 必須要件の充足状況：
  - ☒ 4種類のMLモデル実装完了（Keras, LightGBM, PyCaret, RandomForest）
  - ☒ 予測精度目標達成（テストデータでRMSE=185kW, R2=0.92）
  - ☒ メモリ最適化実装（float32活用で52%削減達成）
  - ☒ 統一アーキテクチャ実装（@dataclass + robust\_model\_operation）
  - ☒ 自動データ取得機能実装（TEPCO API + Open-Meteo API）
  - ☒ 可視化機能実装（16:9統一フォーマット、日本語対応）
- 完成サンプルとの比較結果：
  - 要件定義書との整合性: 100% (全11モジュール要件満足)
  - 性能指標達成度: 108% (目標値を上回る性能達成)
  - 実装完成度: 95% (tomorrow予測拡張機能は今後実装予定)

## 5. 品質評価（自己評価）

- 正確性：★★★★☆ 4つのMLモデルでクロスバリデーション実施、統計的有意性確認済み。予測精度目標値を8%上回る結果を達成。
- 完全性：★★★★☆ 11のモジュール全てが実装完了、統合テスト実施済み。tomorrow予測の自動化機能が一部未完成。
- 一貫性：★★★★★ @dataclass設定管理とrobust\_model\_operationデコレータによる統一アーキテクチャを全モジュールで適用。
- 可読性：★★★★★ PEP 8準拠100%、型ヒント完全実装、docstring記載率100%、モジュール分離による責務明確化。
- 再現性：★★★★★ 固定シード設定(RANDOM\_STATE=42)、環境変数統一管理、依存バージョン固定により同一結果保証。
- 検証可能性：★★★★☆ ユニットテスト80%カバレッジ、パフォーマンス監視ログ、予測結果CSV出力による定量評価可能。
- 保守性：★★★★★ モジュール設計、設定外部化、エラーハンドリング統一、CI/CD対応により長期保守性確保。

## 6. 検証結果

- 検証方法：
  - 単体テスト: pytest使用、各モジュール独立検証
  - 結合テスト: 4モデル統合実行、データフロー検証
  - 性能テスト: 大規模データセット(10,000件)での負荷検証
  - 予測精度検証: ホールドアウト法による未来データ予測精度評価
- 検証環境：
  - Windows 11 Pro (Intel Core i7-8750H, 16GB RAM, NVIDIA GTX 1060)
  - Python 3.10.11, TensorFlow 2.13.0, LightGBM 4.0.0
  - テストデータ: 2024年1-12月実績（8,760時間データ）
- 検証結果概要：
  - 予測精度: RMSE=185kW (目標200kW以下 ☒) , R2=0.92 (目標0.9以上 ☒)
  - 処理性能: メモリ削減52% (目標50% ☒) , 学習時間短縮35% (目標30% ☒)
  - 安定性: 連続実行100回でエラー発生0件、結果再現性100%
  - 可用性: システム稼働率99.7% (目標99.5%以上 ☒)
- 不具合や改善点：
  - 軽微な改善点: tomorrow予測の自動スケジューリング機能未実装
  - パフォーマンス改善余地: GPU活用による更なる高速化可能性
  - 機能拡張案: 異常値検知・アラート機能の追加検討

## 7. 改善提案

- 今後の改善案：
  - 自動化強化**: TEPCO APIからの自動データ取得機能完全実装
  - GPU最適化**: TensorFlow GPU活用による学習時間更なる短縮（目標50%削減）
  - アンサンブル学習**: 4モデルの予測結果を統合する重み付き平均モデル
  - 異常検知**: 予測値が異常範囲を超えた場合の自動アラート機能
  - APIサーバー化**: REST API提供による他システムとの連携機能
  - リアルタイム予測**: ストリーミングデータによるリアルタイム需要予測
- 他工程への展開可能性：
  - ②ワークフロー設計書: MLモデル組み込み型業務フロー設計に応用可能
  - ③経費登録ワークフロー: 予算予測・異常検知モデルとして活用可能
  - ④経費管理アプリ: 予測ダッシュボード機能として統合可能
  - 汎用時系列予測フレームワークとして他ドメイン（売上予測、在庫予測等）への適用可能

## 8. 添付・参考資料

- 添付ファイル名：
  - power\_demand\_models\_source\_code.zip（全ソースコード）
  - model\_performance\_report.xlsx（詳細性能評価結果）
  - prediction\_accuracy\_graphs.pdf（予測精度可視化レポート）
  - memory\_optimization\_analysis.csv（メモリ最適化効果分析）
  - unit\_test\_coverage\_report.html（テストカバレッジレポート）
- 参考URLや文献：
  - TEPCO電力予報: <https://www.tepcoco.jp/forecast/>
  - Open-Meteo Weather API: <https://open-meteo.com/>
  - TensorFlow Time Series Forecasting: [https://www.tensorflow.org/tutorials/structured\\_data/time\\_series](https://www.tensorflow.org/tutorials/structured_data/time_series)
  - LightGBM Parameters Tuning: <https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html>
  - 電力需要予測の機械学習アプローチ: 日本電気学会論文誌参照