

実装タスク: 電力需要予測システム

作成日: 2025年11月26日 バージョン: 1.0.0 ステータス: 全タスク完了 関連ドキュメント: [spec.md](#), [plan.md](#)

実装状況サマリー

カテゴリ	完了	未完了	合計
Phase 1: Setup	8	0	8
Phase 2: Foundational	5	0	5
Phase 3: Core Features	12	0	12
Phase 4: Integration	8	0	8
Phase 5: Testing	5	0	5
Phase 6: Deployment	7	0	7
合計	45	0	45

実装スケジュール (ガントチャート)

電力需要予測システム 実装スケジュール (2025-12-15開始、土日・休日考慮)



Phase 1: Setup (セットアップ)

T001: プロジェクト初期化

- GitHubリポジトリ作成 (<https://github.com/J1921604/Power-Demand-Forecast>)
- README.md作成
- .gitignore設定 (Python, pycache, env除外)
- LICENSE設定 (MIT License)

T002: Python環境構築

- Python 3.10.11インストール確認
- requirements.txt作成 (LightGBM, Keras, scikit-learn, PyCaret, pandas, matplotlib)
- 仮想環境セットアップ (.venv)

T003: ディレクトリ構造作成

- AI/data/ (データ保存)
- AI/train/ (モデル訓練)
- AI/tomorrow/ (翌日予測)
- AI/dashboard/ (Webダッシュボード)
- tests/ (テストコード)
- docs/ (ドキュメント)

T004: データ収集

- 電力需要データ収集 (juyo-2016.csv ~ juyo-2025.csv)
- 気温データ収集 (temperature-2016.csv ~ temperature-2024.csv)
- データ検証 (欠損値・異常値チェック)

T005: 憲法ファイル作成

- .specify/memory/constitution.md 作成
- テスト駆動開発、セキュリティ、パフォーマンス要件定義

T006: テンプレート準備

- .specify/templates/plan-template.md
- .specify/templates/spec-template.md
- .specify/templates/tasks-template.md

T007: 仕様書作成

- specs/001-Power-Demand-Forecast/spec.md (機能仕様書)
- specs/001-Power-Demand-Forecast/requirements.md (要件定義書)

T008: 実装計画作成

- specs/main/plan.md (実装計画)
- specs/main/research.md (リサーチ成果)
- specs/main/data-model.md (データモデル仕様)
- specs/main/contracts/open-meteo-api.yaml (API仕様)
- specs/main/quickstart.md (クイックスタート)

Phase 2: Foundational (基盤実装)

T009: データ前処理パイプライン

- AI/data/data.py 実装 (特徴量生成: MONTH, WEEK, HOUR, TEMP)
- 環境変数AI_TARGET_YEARS設定
- データ結合・分割 (X.csv, Y.csv, Xtrain.csv, Ytrain.csv, Xtest.csv, Ytest.csv)

T010: Open-Meteo API連携

- AI/tomorrow/temp.py 実装 (過去7日+予測7日の気温取得)
- HTTPSリスト実装
- リトライ機構実装 (3回トライ)

T011: モデル訓練基盤

- StandardScaler実装 (標準化)
- モデル保存機構 (pickle, h5形式)
- グラフ描画機構 (16:9比率、日本語対応)

T012: HTTPサーバー実装

- AI/server.py 実装 (ポート8002)
- CORS設定
- /run-data, /run-train, /run-tomorrow エンドポイント

T013: ローカルストレージ永続化

- localStorageによる学習年保存
- モデル別独立記憶 (ai_training_years_[model])

Phase 3: Core Features (コア機能実装)

T014: LightGBMモデル実装

- AI/train/lightgbm/LightGBM_train.py (学習)
- AI/tomorrow/lightgbm/LightGBM_tomorrow.py (翌日予測)
- AI/train/LightGBM/LightGBM_optimize_years.py (組み合わせ検証)
- ハイパラメータ設定 (n_estimators=100, learning_rate=0.1)

T015: Kerasモデル実装

- AI/train/Keras/Keras_train.py (学習)
- AI/tomorrow/Keras/Keras_tomorrow.py (翌日予測)
- AI/train/Keras/Keras_optimize_years.py (組み合わせ検証)
- ニューラルネットワーク設計 (Dense 128-64-32-1)

T016: RandomForestモデル実装

- AI/train/RandomForest/RandomForest_train.py (学習)
- AI/tomorrow/RandomForest/RandomForest_tomorrow.py (翌日予測)
- AI/train/RandomForest/RandomForest_optimize_years.py (組み合わせ検証)
- ハイパラメータ設定 (n_estimators=100)

T017: PyCaretモデル実装

- AI/train/Pycaret/Pycaret_train.py (学習)
- AI/tomorrow/Pycaret/Pycaret_tomorrow.py (翌日予測)
- AI/train/Pycaret/Pycaret_optimize_years.py (組み合わせ検証)
- AutoML設定 (session_id=42, verbose=False)

T018: メトリクス生成

- metrics.json生成 (RMSE, R², MAE)
- 4モデルの精度指標集約

T019: Webダッシュボード実装

- AI/dashboard/index.html (UI設計)
- モデル選択ボタン (4つ)
- 学習年選択ボタン (2016-2024)
- データ処理・学習・予測ボタン
- ネットワーキング・マゼタエフェクト

T020: ダッシュボード起動スクリプト

- start-dashboard.ps1 実装
- Python 3.10検出
- 依存パッケージチェック
- ブラウザ自動起動

Phase 4: Integration (統合実装)

T021: GitHub Actionsワークフロー

- .github/workflows/daily-forecast.yml 作成
- Cronトリガー設定 (0 22 * * * = JST 07:00)
- Push/PRトリガー設定
- workflow_dispatch 実行

T022: CI/CDパイプライン構築

- Python 3.10.11セットアップ
- 依存関係インストール
- 気温データ取得
- データ処理・訓練・予測
- GitHub Pagesデプロイ

T023: R²閾値監視

- R² < 0.8検出ロジック
- GitHub-script@v7を使用

T024: GitHub Pages設定

- index.html作成 (静的ページ)
- actions/deploy-pages@v4統合
- _siteディレクトリ構成

T025: リモートプラン管理

- spec/001-Power-Demand-Forecastブランチマージ

T026: 単体テスト

- tests/unit/test_data.py (data.py検証)
- tests/unit/test_optimize_years.py (組み合わせ検証)

T027: 総合テスト

- tests/integration/test_metrics.py (metrics.json生成検証)
- tests/integration/test_rolling_cv.py (ローリング交差検証)

T028: 契約テスト

- T028契約テスト
- T029 E2Eテスト
- T030 パフォーマンステスト

T029: E2Eテスト

- tests/e2e/test_dashboard.py (Webダッシュボード操作)
- tests/e2e/test_optimize.py (組み合わせ検証)

T030: パフォーマンステスト

- tests/performance/test_training_time.py (訓練時間 ≤ 10秒)
- tests/performance/test_optimize_time.py (検証時間 ≤ 5分)

Phase 5: Testing (テスト実装)

T031: ドキュメント作成

- docs/完全仕様書.md
- docs/使用手順書.md
- docs/DEPLOY_GUIDE.md
- docs/GITHUB_ACTIONS_TEST.md
- docs/IMPLEMENTATION_REPORT.md
- docs/TESTING_GUIDE.md
- docs/RELEASE_NOTES_v1.0.0.md

T032: README更新

- プロジェクト概要
- クックスタート手順
- トラブルシューティング
- リンク修正 (<https://github.com/J1921604/Power-Demand-Forecast>)

T033: バージョン統一

- 全ドキュメントのバージョン: 1.0.0
- 作成日: 更新日: 2025年11月26日

T034: コミット規約準備

- .github/copilot-commit-message-instructions.md 作成

T035: 初回デプロイ

- mainブランチへプッシュ
- GitHub Actions自動実行
- GitHub Pagesデプロイ

T036: 精度検証

- LightGBM R² ≥ 0.90達成確認
- Keras R² ≥ 0.88達成確認
- RandomForest R² ≥ 0.87達成確認
- PyCaret R² ≥ 0.90達成確認

Phase 6: Deployment (デプロイ)

T037: ローカルレビュー検証

- start-dashboard.ps1 実行確認
- ブラウザ自動起動確認
- localStorage永続化確認

完了基準

全体

- 全45タスク完了
- GitHub Actions自動実行成功
- GitHub Pages公開成功
- R² ≥ 0.80達成 (全モデル)

品質

- pytestカバレッジ ≥ 80%
- E2Eテスト100%成功
- ドキュメント網羅性100%

デプロイ

- 毎日JST 07:00自動実行確認
- R² < 0.8時のIssue自動作成確認
- <input checked="" type