# 病院経営分析ダッシュボード 仕様書

## 1. 概要

「病院経営分析ダッシュボード」は、病院の経営指標を可視化・分析するためのウェブアプリケーションです。Streamlitを活用して開発されており、入院患者数や在院日数などの指標を元に、病院経営に必要な意思決定をデータドリブンにサポートします。

**Ver. 2.0での主要更新**

* **統合分析タブ**の新規追加（ALOS分析、曜日別分析、個別分析、予測分析を統合）
* **診療科マッピング機能**の実装
* **メモリ管理システム**の強化
* **バッチ処理最適化**とマルチプロセス対応
* **データ品質監視**機能の追加
* **Parquetキャッシュ**による高速データ処理

## 2. システム構成

### 2.1 技術スタック

* **フレームワーク**: Streamlit 1.29+
* **データ処理**: pandas 2.1+, numpy 1.24+, pyarrow（Parquet処理）
* **可視化**: Plotly 5.17+ (Graph Objects, Express)
* **PDF生成**: ReportLab 4.0+, PyPDF2 3.0+
* **並列処理**: multiprocessing, concurrent.futures, joblib
* **統計分析**: statsmodels, scipy, scikit-learn
* **機械学習**: XGBoost, LightGBM, Prophet（時系列予測）
* **システム監視**: psutil, memory-profiler
* **その他**: jpholiday, python-dateutil, great-expectations

### 2.2 バージョン情報

* **現行バージョン**: Ver. 2.0
* **前バージョンからの互換性**: 完全互換

## 3. アーキテクチャ設計

### 3.1 モジュール構成

app.py # メインアプリケーション

├── data\_processing\_tab.py # データ処理タブ

├── analysis\_tabs.py # 統合分析タブ（新規）

│ ├── alos\_analysis\_tab.py # 平均在院日数分析

│ ├── dow\_analysis\_tab.py # 曜日別分析

│ ├── individual\_analysis\_tab.py # 個別分析

│ └── forecast\_analysis\_tab.py # 予測分析

├── revenue\_dashboard\_tab.py # 収益ダッシュボード

├── department\_tables\_tab.py # 部門別テーブル

└── pdf\_output\_tab.py # PDF出力

### 3.2 データ処理フロー

データ入力 → 前処理 → 統合処理 → 分析処理 → 可視化 → 出力

↓ ↓ ↓ ↓ ↓ ↓

Excel/CSV → 検証 → Parquet → KPI計算 → チャート → PDF

## 4. 主要機能

### 4.1 データ処理タブ

#### 4.1.1 データ入力機能

* **固定ファイル**: 基本となるExcelファイル（.xlsx, .xls）
* **追加ファイル**: 補完データファイル（複数アップロード可）
* **目標値ファイル**: CSV形式の目標設定ファイル
* **Parquetキャッシュ**: 高速再読み込み機能

#### 4.1.2 必須データ列

- 日付（Date型）

- 病棟コード（String）

- 診療科名（String）

- 在院患者数（Integer）

- 入院患者数（Integer）

- 緊急入院患者数（Integer）

- 退院患者数（Integer）

- 死亡患者数（Integer）

#### 4.1.3 データ品質監視

* **重複チェック**: ファイル間重複の効率的検出
* **欠損値処理**: 自動補完と警告表示
* **異常値検出**: 統計的手法による検出
* **データ整合性**: 論理的整合性の検証

### 4.2 経営ダッシュボード

#### 4.2.1 収益管理セクション

* **延べ在院日数（人日）**:
  + 目標達成率（カラーコード付き）
  + 前月比・前年比表示
  + トレンド分析
* **新入院患者数**:
  + 目標達成率監視
  + 診療科別内訳
* **推計収益**:
  + 延べ在院日数 × 平均入院料単価
  + 診療科別収益構成比
  + 月別収益推移

#### 4.2.2 運営指標セクション

* **平均在院日数（ALOS）**:
  + 前月比変化率
  + 閾値ベースのステータス表示
  + 診療科別比較
* **病床利用率**:
  + リアルタイム監視
  + 目標値との比較
  + 稼働効率分析
* **病床回転数**:
  + 効率性指標
  + ベンチマーク比較

#### 4.2.3 アラート機能

* **閾値監視**: 設定値の85%-115%で自動アラート
* **トレンド監視**: 急激な変化の検出
* **収益影響分析**: 指標変化の収益影響度

### 4.3 統合分析タブ（新機能）

#### 4.3.1 ALOS分析

* **集計粒度**: 月単位/週単位
* **集計単位**: 病院全体/病棟別/診療科別
* **移動平均**: 7日、14日、30日移動平均
* **ベンチマーク比較**: 目標値との比較分析

#### 4.3.2 曜日別分析

* **パターン分析**: 曜日別入退院パターン
* **ヒートマップ**: 診療科×曜日のマトリックス
* **季節性分析**: 月別・四半期別パターン
* **予測活用**: 曜日パターンを活用した需要予測

#### 4.3.3 個別分析

* **カスタム期間**: ユーザー定義期間での分析
* **多次元分析**: 複数指標の同時分析
* **ドリルダウン**: 詳細レベルでの分析
* **比較分析**: 期間比較、部門比較

#### 4.3.4 予測分析

* **複数モデル**:
  + 単純移動平均
  + Holt-Winters（季節性考慮）
  + ARIMA（自己回帰）
  + Prophet（Facebook開発）
  + XGBoost（機械学習）
* **精度評価**: MAE、RMSE、MAPE
* **信頼区間**: 予測の不確実性表示
* **シナリオ分析**: What-if分析

### 4.4 部門別テーブル機能

#### 4.4.1 病棟別テーブル

* **基本情報モード**:
  + 在院患者数、入退院患者数
  + 平均在院日数、病床利用率
* **詳細情報モード**:
  + 緊急入院率、死亡率
  + 病床回転数、収益貢献度
* **診療科マッピング**: 表示名の自動変換

#### 4.4.2 診療科別テーブル

* **フィルタリング機能**:
  + 全診療科表示
  + 主要診療科のみ
  + カスタム選択
* **並び替え機能**:
  + 患者数順、達成率順
  + カスタムソート
* **集計機能**:
  + 小計・合計の自動計算
  + 構成比表示

### 4.5 PDF出力機能

#### 4.5.1 個別PDF出力

* **レイアウト**: 縦向き/横向き選択
* **内容構成**:
  + エグゼクティブサマリー
  + トレンドグラフ
  + 予測データ
  + KPIテーブル
* **カスタマイズ**: テンプレート選択可能

#### 4.5.2 一括PDF出力

* **並列処理**: マルチプロセス対応
* **進捗表示**: リアルタイム進捗
* **エラーハンドリング**: 部分失敗でも継続
* **圧縮出力**: ZIP形式での一括ダウンロード

## 5. データ処理仕様

### 5.1 前処理フロー

# 前処理パイプライン

1. ファイル読み込み（Excel/CSV）

2. 列名標準化・データ型変換

3. 欠損値処理・異常値検出

4. 重複除去（効率的アルゴリズム）

5. 派生指標計算

6. 平日/休日フラグ付与

7. 診療科マッピング適用

8. Parquetキャッシュ保存

### 5.2 KPI計算式

#### 5.2.1 基本指標

* **平均在院日数（ALOS）**:
* 期間延べ在院患者数 ÷ ((入院患者数 + 退院患者数) ÷ 2)
* **病床利用率**:
* (日平均在院患者数 ÷ 総病床数) × 100
* **病床回転数**:
* 期間退院患者数 ÷ 期間平均在院患者数

#### 5.2.2 収益指標

* **延べ在院日数（人日）**:
* Σ(日別在院患者数)
* **推計収益**:
* 延べ在院日数 × 平均入院料単価
* **収益効率**:
* 推計収益 ÷ 利用可能病床日数

#### 5.2.3 品質指標

* **緊急入院率**:
* (緊急入院患者数 ÷ 総入院患者数) × 100
* **死亡率**:
* (死亡患者数 ÷ 総退院患者数) × 100

### 5.3 予測モデル仕様

#### 5.3.1 時系列モデル

* **ARIMA**: 自己回帰統合移動平均モデル
* **Prophet**: トレンド・季節性を自動検出
* **Holt-Winters**: 指数平滑法（加法・乗法）

#### 5.3.2 機械学習モデル

* **XGBoost**: 勾配ブースティング
* **LightGBM**: 軽量勾配ブースティング
* **Random Forest**: アンサンブル学習

## 6. システム仕様

### 6.1 パフォーマンス要件

#### 6.1.1 処理速度

* **データ読み込み**: 100MB以下のファイルで30秒以内
* **グラフ生成**: 5秒以内
* **PDF生成**: 個別1分以内、一括10分以内
* **予測計算**: 3年分データで2分以内

#### 6.1.2 メモリ管理

* **使用量監視**: リアルタイム監視
* **自動クリーンアップ**: 85%超過時自動実行
* **キャッシュ管理**: LRU方式での効率的管理

### 6.2 システム要件

#### 6.2.1 最小要件

* **CPU**: 2コア以上
* **メモリ**: 4GB以上
* **ストレージ**: 2GB以上の空き容量
* **OS**: Windows 10+, macOS 10.15+, Ubuntu 18.04+

#### 6.2.2 推奨要件

* **CPU**: 4コア以上
* **メモリ**: 8GB以上
* **ストレージ**: 10GB以上の空き容量
* **GPU**: CUDA対応（機械学習加速用、オプション）

#### 6.2.3 サポートブラウザ

* **Chrome**: 90.0+
* **Firefox**: 88.0+
* **Safari**: 14.0+
* **Edge**: 90.0+

## 7. セキュリティ仕様

### 7.1 データ保護

* **ファイル暗号化**: アップロードファイルの暗号化保存
* **一時ファイル管理**: 自動削除とセキュア削除
* **メモリ保護**: 機密データのメモリクリア

### 7.2 アクセス制御

* **ロールベース**: 閲覧者・分析者・管理者
* **セッション管理**: タイムアウト機能
* **監査ログ**: アクセスログの記録

## 8. 目標値ファイル仕様

### 8.1 CSV形式仕様

部門コード,部門名,区分,目標値,延べ在院日数目標,新入院患者数目標,平均入院料単価,病床数

全体,病院全体,全日,500,16500,120,55000,612

01A,内科A病棟,全日,80,2400,30,52000,50

02B,外科B病棟,全日,60,1800,25,58000,40

内科,内科,全日,150,4500,45,53000,

外科,外科,全日,120,3600,35,59000,

### 8.2 項目説明

* **部門コード**: 病棟コードまたは診療科コード
* **部門名**: 表示用名称
* **区分**: 全日/平日/休日（現在は全日のみ対応）
* **目標値**: 基本目標値（患者数）
* **延べ在院日数目標**: 月間目標人日数
* **新入院患者数目標**: 月間目標新入院患者数
* **平均入院料単価**: 部門別単価設定
* **病床数**: 部門別病床数（オプション）

## 9. 運用ガイドライン

### 9.1 日常運用

#### 9.1.1 データ更新頻度

* **日次更新**: 前日データの取り込み
* **週次レビュー**: 週間サマリーの確認
* **月次分析**: 月間実績と目標の比較分析

#### 9.1.2 監視項目

* **システムパフォーマンス**: CPU・メモリ使用率
* **データ品質**: 欠損・異常値の監視
* **予測精度**: 実績との乖離監視

### 9.2 トラブルシューティング

#### 9.2.1 一般的な問題

* **ファイル読み込みエラー**: 形式・エンコーディング確認
* **メモリ不足**: データサイズ削減・分割処理
* **予測精度低下**: モデル再学習・パラメータ調整

#### 9.2.2 復旧手順

* **データバックアップ**: 定期的なデータ保存
* **設定リセット**: 初期設定への復旧
* **ログ分析**: エラーログの詳細分析

## 10. 今後の拡張計画

### 10.1 短期計画（3-6ヶ月）

* **リアルタイム連携**: EHRシステムとの直接連携
* **モバイル対応**: レスポンシブデザインの強化
* **多言語対応**: 英語インターフェースの追加

### 10.2 中期計画（6-12ヶ月）

* **AI予測強化**: 深層学習モデルの導入
* **自動レポート**: 定期レポートの自動生成・配信
* **アラート連携**: メール・Slack通知機能

### 10.3 長期計画（1-2年）

* **マルチテナント**: 複数病院対応
* **クラウド展開**: AWS・Azure対応
* **API提供**: 外部システム連携API

## 11. 品質保証

### 11.1 テスト仕様

* **単体テスト**: 各関数・モジュールのテスト
* **結合テスト**: モジュール間の連携テスト
* **パフォーマンステスト**: 負荷・速度テスト
* **ユーザビリティテスト**: 実際の使用シーンでのテスト

### 11.2 品質メトリクス

* **コードカバレッジ**: 80%以上
* **バグ検出率**: リリース前90%以上
* **ユーザー満足度**: 4.0/5.0以上（アンケート）
* **システム可用性**: 99.5%以上

本仕様書は、システムの継続的な改善により定期的に更新されます。 最新版は開発チームにお問い合わせください。