MCP Server Documentation

Clinical Data Analysis System
Heart Disease Dataset - UCI Repository

CIRILLO MICHELE

February 2, 2026

Abstract

This document provides a complete guide to the three MCP (Model Context Protocol) servers implemented for clinical data analysis on the Heart Disease dataset from the UCI Machine Learning Repository. The servers offer ETL functionalities, data analysis, and knowledge graph construction to support advanced clinical analysis through Claude AI.

Contents

1 Introduction **3**
1.1 Dataset: Heart Disease UCI . . . 3

2 MCP Server 1: ClinicalDataETL **3**
2.1 Description . . . 3
2.2 Available Tools . . . 3
2.3 Feature Engineering . . . 4

3 MCP Server 2: ClinicalDataAnalyzer **5**
3.1 Description . . . 5
3.2 Available Tools . . . 5
3.3 Capabilities . . . 6

4 MCP Server 3: MedicalKnowledgeGraph **6**
4.1 Description . . . 6
4.2 Available Tools . . . 7
4.3 Knowledge Graph Architecture . . . 8

5 Integration with Claude AI **8**
5.1 Configuration . . . 8
5.2 Typical Analysis Workflow . . . 9
5.3 Best Practices . . . 9
5.4 Effective Prompts . . . 10

6 Conclusions **10**
6.1 System Advantages . . . 10
6.2 Future Developments . . . 11

1 Introduction

The system consists of three interconnected MCP servers that work synergistically to provide a complete clinical data analysis infrastructure:

1. **ClinicalDataETL**: Server dedicated to extraction, transformation, and loading of data from the UCI repository
2. **ClinicalDataAnalyzer**: Statistical and exploratory analysis engine for clinical data
3. **MedicalKnowledgeGraph**: Graph-based knowledge representation system to identify clinical relationships

1.1 Dataset: Heart Disease UCI

The dataset used comes from the UCI Machine Learning Repository and contains clinical data of 303 patients collected at the Cleveland Clinic. It includes 14 clinical variables that allow prediction of the presence of heart disease.

Main characteristics:

- 303 patient records
- 14 original clinical features
- Binary target feature (presence/absence of heart disease)
- Public and anonymized data

2 MCP Server 1: ClinicalDataETL

2.1 Description

The ClinicalDataETL server is responsible for managing the data lifecycle: downloading from the UCI repository, cleaning, transformation, and enrichment through feature engineering.

2.2 Available Tools

fetch_and_process_heart_data

Description: Downloads and processes the Heart Disease dataset from the UCI Repository.

Parameters:

- `sample_size` (optional): Number of records to sample. If `None`, the entire dataset is used.

Output: Processed data with feature engineering and descriptive statistics.

Functionalities:

- Automatic download from UCI
- Missing value handling (represented as '?')
- Data type conversion
- Creation of derived features

get_dataset_info

Description: Returns detailed information about the available dataset.
Parameters: None
Output: Complete dataset metadata including:

- Original columns and added features
- Source URL
- Description and technical notes
- Usage instructions

analyze_heart_disease

Description: Performs an in-depth analysis of the dataset with statistics, correlations, and clinical insights.
Parameters:

- `sample_size` (optional): Sample size to analyze

Output:

- Complete descriptive statistics
- Correlation matrix
- Automatic clinical insights
- Distribution by age, sex, chest pain type

export_processed_data

Description: Exports processed data in different formats.
Parameters:

- `format`: Export format (`json`, `csv`, `dict`)
- `sample_size` (optional): Number of records to export

Output: Data in the requested format ready for external use.

2.3 Feature Engineering

The server automatically enriches the dataset with the following derived features:

- **age_group**: Age categorization (young/adult/elderly)
- **bp_category**: Blood pressure category (normal/elevated/high)
- **chol_category**: Cholesterol level (normal/borderline/high)
- **risk_score**: Calculated cardiovascular risk score
- **has_disease**: Binary disease presence flag

3 MCP Server 2: ClinicalDataAnalyzer

3.1 Description

ClinicalDataAnalyzer provides advanced statistical and exploratory analysis functionalities specifically designed for clinical data. It implements anomaly detection algorithms, correlation analysis, and data quality evaluation.

3.2 Available Tools

analyze_clinical_data

Description: Preliminary clinical data analysis with descriptive statistics.
Parameters:

- `sample_size` (optional): Number of records to sample for analysis

Output: Complete descriptive statistics for all variables:

- Mean, median, standard deviation
- Minimum and maximum values
- Quartiles and interquartile range
- Counts and frequencies for categorical variables

exploratory_data_analysis

Description: Complete exploratory data analysis (EDA) of the clinical dataset.
Parameters:

- `sample_size` (optional): Sample size

Output: Complete report including:

- Advanced descriptive statistics
- Correlation analysis between variables
- Pattern and trend identification
- Target variable distribution
- Relationships between features and clinical outcomes

detect_data_issues

Description: Automatic detection of problems in clinical data.
Parameters:

- `sample_size` (optional): Sample to analyze

Output: Data quality report with:

- Missing value identification
- Statistical outlier detection
- Identification of anomalous values in clinical context
- Inconsistencies between variables
- Recommendations for data cleaning

get_dataset_info

Description: Information about the public dataset in use.
Parameters: None
Output: Dataset metadata and description.

get_analyzer_info

Description: Information about the clinical data analysis system.
Parameters: None
Output: Version, capabilities, and operations supported by the analyzer.

3.3 Capabilities

The ClinicalDataAnalyzer server offers the following capabilities:

1. **Preliminary data analysis**: Quick overview of main characteristics
2. **Complete exploratory analysis**: Deep dive into distributions and relationships
3. **Data issue detection**: Automatic quality assurance
4. **Descriptive statistics**: Complete metrics for each variable
5. **Correlation analysis**: Identification of relationships between clinical variables
6. **Outlier detection**: Identification of anomalous values using statistical methods

4 MCP Server 3: MedicalKnowledgeGraph

4.1 Description

MedicalKnowledgeGraph builds and manages a knowledge graph that represents complex relationships between clinical variables. It uses graph analytics techniques to identify hidden patterns and non-obvious connections in medical data.

4.2 Available Tools

explore_relations

Description: Explores how a feature is connected to others in the Knowledge Graph.
Parameters:

- `feature`: Name of the feature to explore (default: `target`)

Output: Relationship analysis including:

- List of most correlated features
- Relationship strength (edge weight)
- Type of correlation (positive/negative)
- Clinical interpretation of relationships
- Suggestions for further investigation

Explorable features: `age`, `sex`, `cp`, `trestbps`, `chol`, `fbs`, `restecg`, `thalach`, `exang`, `oldpeak`, `slope`, `ca`, `thal`, `target`

clinical_insights

Description: Generates essential clinical insights from the Knowledge Graph.
Parameters: None
Output: Complete report with:

- Top 10 strongest relationships in the dataset
- Clinical interpretation of each relationship
- Identified significant patterns
- Main risk factors for heart disease
- Identified protective variables
- Recommendations for further analysis

kg_status

Description: Returns the current status of the Knowledge Graph.
Parameters: None
Output:

- System status (data loaded or not)
- Dataset size
- Number of nodes and relationships in the graph
- Available tools
- Simple usage example

4.3 Knowledge Graph Architecture

The knowledge graph is structured as follows:

Nodes: Each clinical feature of the dataset represents a node.

Edges: Relationships between features are represented by weighted edges, where the weight indicates the strength of the correlation (calculated through Pearson correlation for continuous variables and appropriate measures for categorical variables).

Node properties:

- Feature name
- Data type (continuous/categorical)
- Descriptive statistics
- Clinical relevance

Edge properties:

- Correlation coefficient
- Significance p-value
- Clinical interpretation

5 Integration with Claude AI

5.1 Configuration

To use these MCP servers with Claude, they must be configured in Claude's MCP configuration file. The servers must be started and made accessible through the MCP protocol.

5.2 Typical Analysis Workflow

Usage example

Scenario 1: Complete Exploratory Analysis

1. **Data Loading**

   Prompt: "Load the heart disease dataset and give me an overview"

   Claude uses: `ClinicalDataETL:fetch_and_process_heart_data()`

2. **Preliminary Analysis**

   Prompt: "Analyze the data and identify any problems"

   Claude uses:

   - `ClinicalDataAnalyzer:analyze_clinical_data()`
   - `ClinicalDataAnalyzer:detect_data_issues()`

3. **Deep Exploration**

   Prompt: "Give me a complete exploratory analysis"

   Claude uses: `ClinicalDataAnalyzer:exploratory_data_analysis()`

4. **Knowledge Graph Construction**

   Prompt: "Build a knowledge graph and show me the main insights"

   Claude uses:

   - `MedicalKnowledgeGraph:load_heart_disease_data()`
   - `MedicalKnowledgeGraph:clinical_insights()`

Usage example

Scenario 2: Specific Investigation

1. *Prompt:* "What are the main factors correlated with cholesterol?"

   Claude uses:

   - `MedicalKnowledgeGraph:load_heart_disease_data()`
   - `MedicalKnowledgeGraph:explore_relations(feature="chol")`

2. *Prompt:* "Analyze the relationship between age and presence of heart disease"

   Claude uses:

   - `ClinicalDataAnalyzer:exploratory_data_analysis()`
   - `MedicalKnowledgeGraph:explore_relations(feature="age")`
   - `MedicalKnowledgeGraph:explore_relations(feature="target")`

5.3 Best Practices

1. **Start with loading**: Always first use `fetch_and_process_heart_data()` or `load_heart_disease_data(`
   to prepare the data.

2. **Check quality**: Run `detect_data_issues()` before deep analysis to identify potential problems.
3. **Leverage the Knowledge Graph**: For questions about relationships between variables, the Knowledge Graph provides richer insights compared to simple correlations.
4. **Sampling**: For quick tests, use the `sample_size` parameter to work on data subsets.
5. **Export**: Use `export_processed_data()` if you need processed data for external analysis.

5.4 Effective Prompts

For general analysis:

- "Load and analyze the heart disease dataset"
- "Give me a complete overview of clinical data"
- "Identify the main cardiovascular risk factors"

For specific investigations:

- "Explore cholesterol relationships with other variables"
- "Which features are most correlated with disease presence?"
- "Analyze the impact of age on cardiovascular parameters"

For quality assurance:

- "Check data quality and identify problems"
- "Are there outliers or anomalous values in the dataset?"
- "Analyze data completeness"

6 Conclusions

The three MCP servers provide a complete infrastructure for clinical data analysis through Claude AI. The combination of robust ETL, advanced statistical analysis, and knowledge graph enables deep insights into cardiovascular risk factors and relationships between clinical variables.

6.1 System Advantages

- **Automation**: Automatic management of the complete data lifecycle
- **Quality**: Automatic detection of problems and anomalies
- **Insights**: Knowledge graph to discover non-obvious relationships
- **Flexibility**: Sampling and export in various formats
- **Integration**: Seamless integration with Claude's analytical capabilities

6.2 Future Developments

Possible system extensions include:

- Support for other UCI datasets
- Integrated machine learning algorithms
- Interactive knowledge graph visualizations
- Export in standard medical formats (FHIR, HL7)
- Integration with clinical decision support systems