WellRithms Take-Home for Kiril Simov

WellRithms MLE Take Home Assessment

Author: Kiril Simov (GitHub: krs2022)

This repository contains solutions for the WellRithms Machine Learning Engineer take-home assessment, featuring advanced clustering analysis and PDF table extraction with OCR processing.

Directory Structure

```
assignment_krs/
  README.MD                             # This documentation file
  task1/                                # Clustering analysis task
      task1_clustering.py              # Main clustering script
      task1_output.txt                 # Console output from script execution
      task1_clustering_results.csv     # Processed data with cluster assignments
      clustering_results_*.png         # Visualization plots (3 files)
      WellRithms_Text_MLE Take Home Assessment.xlsx  # Input data
  task2/                                # PDF table extraction task
      task2_statements.py              # Main table extraction script
      test_basic_functionality.py      # Unit tests for core functions
      task2_table_*_results.csv        # Extracted table data (3 files)
      debug_column_overlay_*.png       # Column detection visualization (3 files)
      WellRithms MLE Take Home Assessment.pdf  # Input PDF document
```

Quick Start

Prerequisites

Ensure you have Python 3.7+ installed with the following packages:

For Task 1 (Clustering):

```
pip install pandas numpy matplotlib scikit-learn openpyxl
```

For Task 2 (PDF Processing):

```
pip install pandas numpy opencv-python pillow pytesseract PyPDF2 PyMuPDF
```

Note: You'll also need to install Tesseract OCR on your system:

- **macOS:** `brew install tesseract`
- **Ubuntu:** `sudo apt-get install tesseract-ocr`
- **Windows:** Download from GitHub releases

Task 1: Advanced Clustering Analysis

Description

Performs comprehensive clustering analysis on medical equipment data using three different algorithms:

- **K-Means**: Spherical cluster detection with automatic optimization
- **DBSCAN**: Density-based clustering for natural groupings
- **Hierarchical**: Agglomerative clustering for hierarchical relationships

Key Features

- TF-IDF vectorization with n-gram analysis
- Automatic optimal cluster number detection
- Silhouette score evaluation
- t-SNE visualization of results
- Confidence scoring for cluster assignments

Running Task 1

```
cd task1/
python task1_clustering.py
```

Task 1 Output

- **Console**: Progress updates and cluster statistics
- **CSV File**: `task1_clustering_results.csv` with cluster assignments
- **Visualizations**: Three PNG files showing clustering results
- **Processing Time**: ~30-60 seconds for 8,750 records

Output Files Description

- `task1_clustering_results.csv`: Original data + cluster assignments for all three methods + confidence scores
- `clustering_results_kmeans_cluster.png`: K-means clustering visualization
- `clustering_results_dbscan_cluster.png`: DBSCAN clustering visualization
- `clustering_results_hierarchical_cluster.png`: Hierarchical clustering visualization

Task 2: PDF Table Extraction with OCR

Overview

Extracts and processes tables from the last 3 pages of a PDF document using:

- Advanced image preprocessing
- Intelligent column boundary detection
- OCR text extraction with Tesseract
- Data cleaning and validation
- Medical billing data pattern recognition

Capabilities

- PDF image extraction from specific pages
- Dual-method column detection (visual + text-based)
- OCR preprocessing for better accuracy
- Pattern-based data extraction (revenue codes, HCPCS codes, dates, charges)
- Column overlay generation for debugging
- Comprehensive data cleaning and validation

Running Task 2

```
cd task2/
python task2_statements.py
```

Running Tests

```
cd task2/
python test_basic_functionality.py
```

Task 2 Output

- **Console**: Processing progress for each image
- **CSV Files**: Three files (`task2_table_1_results.csv`, etc.) with extracted table data
- **Debug Images**: Column overlay images showing detected boundaries
- **Processing Time**: ~10-30 seconds per image

Generated Files

- `task2_table_*_results.csv`: Structured table data with columns:
  - `REV_CODE`: Revenue codes (4-digit)
  - `DESCRIPTION`: Service descriptions
  - `HCPCS_CODE`: Healthcare procedure codes (5-character)
  - `SERVICE_DATE`: Service dates (parsed to datetime)
  - `UNITS`: Number of units
  - `CHARGES`: Monetary charges (numeric)

- debug_column_overlay_*.png: Visual debugging showing detected column boundaries

Testing

Task 2 includes comprehensive unit tests:

```
cd task2/
python test_basic_functionality.py
```

Tests cover:

- Image preprocessing functionality
- Column boundary detection
- Overlay image creation
- Data cleaning and validation

Performance Notes

- Task 1 uses t-SNE for visualization, which can be computationally intensive
- Task 2 OCR processing depends on image quality and complexity
- Column detection uses multiple algorithms for robustness, included visual overlay for debugging, and unit tests are documented above

Results Summary

Task 1 Results:

- Successfully clustered 8,750 medical equipment items
- K-Means: 49 optimal clusters
- DBSCAN: 330 density-based clusters

- Hierarchical: 49 hierarchical clusters
- Generated confidence scores for cluster assignments

Task 2 Results:

- Extracted 3 tables from PDF images
- Identified columns: REV_CODE, DESCRIPTION, HCPCS_CODE, SERVICE_DATE, UNITS, CHARGES
- Applied OCR error correction and data validation
- Generated debugging visualizations for column detection

Notes

- All scripts include comprehensive error handling and progress reporting
- Output files are automatically saved in respective task directories
- Visualization files help validate clustering and column detection results