Smart Header Detection Implementation Summary

Branch: fix/smart-header-detection

Commit: c3eec68f4ef0eb6541793d5cb18b1665c3e08242
Status: ✅ Pushed to GitHub
Repository: https://github.com/CliveCaseley/soldcomp-analyser2

Problem Statement

The CSV parser was failing to handle files where:
- Row 1 contained TARGET metadata instead of headers
- Headers appeared on row 3 or later (within first 10 rows)
- Files had varying structures with metadata, empty rows, or formatting rows before the actual headers

Example file structure:

```
Row 1: "TARGET is Hawthorns, Horncastle Road, Goulceby, LOUTH LN11 9WB",,,,,1152,,,
Row 2: ,,,,,,,,,,,
Row 3: Date,Address,Postcode,Type,Tenure,Age at sale,Price,Sq. ft,£/
sqft,Bedrooms,Distance,URL
Row 4: 15/05/2025,"Badger's Holt, Horncastle Road, Goulceby",LN11 9WB,Detached
house,Freehold,...
```

Solution Implemented

1. Smart Header Row Detection (`detectHeaders` function)

Key improvements:
- ✅ Scans first 10 rows instead of assuming row 1 is the header
- ✅ Uses fuzzy matching with scoring algorithm to identify the best header row
- ✅ Prioritizes rows containing core columns: `date`, `address`, `postcode`, `price`
- ✅ Requires at least 2 core columns matched before considering a row as header
- ✅ Exact match detection (100% priority) before fuzzy matching (70% threshold)

Scoring algorithm:

```
Row Score = (Number of matched columns) + (Core columns matched × 2)
```

Core column detection:
- A row must match at least 2 of: `date`, `address`, `postcode`, `price`
- This prevents false positives from data rows

2. Enhanced Column Mapping

Header variations supported:
- Date: `date`, `sale date`, `sold date`, `transaction date`
- Address: `address`, `property address`, `full address`
- Postcode: `postcode`, `post code`, `postal code`
- Price: `price`, `sale price`, `sold price`, `amount`
- Type: `type`, `property type`, `house type`
- And many more...

Special handling:
- ✅ `£/sqft` checked BEFORE `Sq. ft` to prevent mismatches
- ✅ URL detection prevents URLs from being mapped to non-URL columns
- ✅ Preserves all standard headers with empty values for unmapped columns

3. Smart Data Normalization (`normalizeData` function)

Updates:
- ✅ Starts processing from row AFTER the detected header row
- ✅ Skips all metadata rows (TARGET info, empty rows, etc.)
- ✅ Properly handles URL-only rows
- ✅ Prevents data loss from rows before headers

Test Results

Test 1: `data (3).csv` - TARGET Metadata on Row 1

File structure:

```
Row 0: ,"TARGET is Hawthorns, Horncastle Road, Goulceby, LOUTH  LN11 9WB",...
Row 1: ,,,,,,,,,,, (empty)
Row 2: Date,Address,Postcode,Type,Tenure,Age at sale,Price,Sq. ft,£/
sqft,Bedrooms,Distance,URL
Row 3: 15/05/2025,"Badger's Holt, Horncastle Road, Goulceby",LN11 9WB,...
```

Parser output:

```
✓ Header row detected at row 2
✓ Matched 12 columns (4 core columns)
✓ Processed 50 data rows (starting from row 3)
✓ No data loss
```

Sample parsed data:

```
{
  "Date of sale": "15/05/2025",
  "Address": "Badger's Holt, Horncastle Road, Goulceby",
  "Postcode": "LN11 9WB",
  "Type": "Detached house",
  "Tenure": "Freehold",
  "Price": "£315,000",
  "Sq. ft": "1130",
  "£/sqft": "£279",
  "Distance": "0.00mi",
  "URL": "https://propertydata.co.uk/transaction/38EDC0C3-0B84-0F63-E063-4704A8C00424"
}
```


Test 2: `output (22).csv` - Standard Format

File structure:

```
Row 0: Date of sale,Address,Postcode,Type,Tenure,Age at sale,Price,...
Row 1: ,EPC Lookup,DN9 3PT,,,,,,,,,,https://...
Row 2: ,7 Fernbank Close,DN9 3PT,,,,,,,,,,,,,,,,,,,,1,,
Row 3: 02 Jul 2025,"Pembroke House, Blakewood Drive, Blaxton, Doncaster DN9 3GX",DN9
3GX,...
```

Parser output:

```
✓ Header row detected at row 0
✓ Matched 24 columns (4 core columns)
✓ Processed 20 data rows (starting from row 1)
✓ Correctly handled EPC lookup and target marker rows
```

Code Changes

File modified: `src/utils/csvParser.js`

Key changes:

1. **detectHeaders() function** (lines 132-249)
- Added row scanning loop (first 10 rows)
- Implemented scoring algorithm
- Added core column requirement check
- Added exact match detection with 100% priority

1. **normalizeData() function** (lines 268-332)
   - Added `headerRowIndex` parameter
   - Changed start index to `headerRowIndex + 1`
   - Added logging for skipped rows

2. **HEADER_VARIATIONS object** (lines 44-67)
   - Reordered entries to check specific headers first
   - Added comment explaining priority

Impact

✅ **Robustness:** Handles varying CSV structures without manual intervention
✅ **Accuracy:** 100% correct header detection across all test files
✅ **No Data Loss:** All data rows properly parsed, metadata properly skipped
✅ **Backward Compatible:** Still works with standard CSV files (headers on row 1)
✅ **Well Logged:** Detailed logging shows exact header detection process

How to Create Pull Request

1. Visit: https://github.com/CliveCaseley/soldcomp-analyser2/compare/fix/smart-header-detection
2. Click “Create pull request”
3. Title: “Fix: Implement smart header detection for messy CSV files”
4. Description: Link to this summary document
5. Submit for review

Pull Request URL

https://github.com/CliveCaseley/soldcomp-analyser2/compare/master...fix/smart-header-detection

---

Implementation Date: December 4, 2025
Author: Clive Caseley
Branch: fix/smart-header-detection
Commit: c3eec68