Scout Problem Solution

I will focus on Joining Datasets, as it presents an interesting challenge in reconciling overlapping agent data across the datasets and integrating it with transactional history. Here's the plan:

Approach to Joining Datasets

**Step 1:** Understanding the Data, Load and inspect all provided datasets (agents.csv, agents\_2.csv, transactions.csv). Identify key columns for joining (e.g., agent IDs, names, brokerage details). Check for overlapping columns and differences in structure.

**Step 2:** Data Cleaning & Preprocessing & standardize Key Fields:

Normalize agent names and brokerage names (e.g., consistent capitalization, removing extra spaces, handling special characters). Standardize date formats (if applicable).

**Deduplicate Records:**

Identify duplicates based on unique identifiers (e.g., agent ID or a composite key like name + contact details). Resolve discrepancies using rules (e.g., prioritize data from the most comprehensive dataset).

**Handle Missing Data:**

Fill missing agent IDs or brokerage data where possible using cross-referencing. Mark records with critical missing data for review.

**Step 3:** Joining Logic

Agent Matching: Match agents across agents.csv, agents\_2.csv, and transactions.csv using:

Unique IDs (if available). Fuzzy matching on names, email, or contact numbers (in case of partial data). Cross-referencing with all\_agents.csv to fill gaps.

**Brokerage Matching:**

Normalize variations in brokerage names (e.g., "Acme Realty" vs. "Acme Real Estate").

Create a mapping table for consistent reference.

**Transaction Linking:**

Join transactions with the unified agent dataset based on agent ID or name.

Ensure transactional records are tagged with the correct agent and brokerage.

**Step 4:** Create a Unified View

Combine all datasets into a single table with normalized agent and brokerage details and linked transactional data. Key fields to include:

Agent ID, Name, Contact Info, Brokerage, Team (if available).

Transaction Details: Property ID, Sale Price, Date, Brokerage.

Dataset Observations

Agents Dataset (agents.csv):

Contains detailed agent information (e.g., names, licenses, contact details, brokerage).

Appears relatively clean but has some columns with missing data.

Agents 2 Dataset (agents\_2.csv):

Similar structure but includes additional fields like MLS details and display addresses.

Includes variations in data for the same agents (e.g., different formats for addresses).

Transactions Dataset (transactions.csv):

Large transactional dataset with property details and agent/brokerage participation.

Many fields, but key identifiers (e.g., agent\_id) are frequently missing.

Next Steps: Joining Datasets

**Step 1:** Identify and Resolve Key Issues

Agent Matching:

Both agents.csv and agents\_2.csv overlap but may have inconsistent data (e.g., names, phone numbers).

Use fuzzy matching on names, emails, or phone numbers when exact matches are unavailable.

Brokerage Matching: Variations in brokerage names across datasets require normalization (e.g., "BHHS" vs. "Berkshire Hathaway").

Transaction Linking: Many transactional records lack explicit agent\_id.

Will rely on joining through agent names and additional details like brokerage affiliation.

**Step 2:** Implementation Plan

Create a unified agent table by merging agents.csv and agents\_2.csv.

Standardize brokerage and agent names using string normalization and fuzzy matching.

Join transactional data to the unified agent table based on matching rules. Let’s start by combining agents.csv and agents\_2.csv into a unified agent table. ​​

**Match Summary:**

Left-only (agents.csv): 993 records unmatched in agents\_2.csv.

Right-only (agents\_2.csv): 993 records unmatched in agents.csv.

Both: 7 records matched by normalized names.

**Observations:**

A large number of records did not match directly, likely due to: Variations in formatting (e.g., middle names or initials).

Missing or incomplete entries in either dataset.

Next Steps: Enhanced Matching

Fuzzy Matching: Use fuzzy logic to identify likely matches for left\_only and right\_only entries.

Score matches based on similarity thresholds (e.g., name + contact).

Manual Inspection for Ambiguities:

Validate matches with high fuzziness scores but conflicting data. Combine Matched and Unmatched Records: Consolidate matched data into a unified table.

Retain unmatched records separately for review or manual intervention.

ETL Approach

**1. Extract**

Load datasets (agents, agents\_2, transactions) from Excel files into Python or a database.

Use libraries such as pandas in Python or SQL LOAD DATA commands for this.

**2. Transform**

Data Cleaning

Normalize Agent Names:

Convert names to lowercase.

Remove extra spaces, special characters, or abbreviations.

Handle Missing Values: Fill critical missing data fields, e.g., missing agent\_id, by matching based on name, email, or phone number. Flag unresolvable cases for manual review.

Standardize Brokerage Names: Create a mapping table to normalize variations (e.g., "BHHS" → "Berkshire Hathaway").

Resolve Duplicates: Identify duplicates using unique keys (e.g., agent name + phone or email).

Consolidate data, keeping the most complete record.

Fuzzy Matching:

Match agents in agents and agents\_2 using fuzzy logic for close name variations.

Joining Datasets

Merge agents and agents\_2 into a unified agent table.

Join transactional data using agent names, agent\_id, or a composite key like name + brokerage.

**3. Load**

Store the final cleaned and unified dataset in a database. Use a structured schema for easy querying:

Agents table: Normalized data for agents and brokers.

Transactions table: Linked to agents and teams.