Geocoding Truck Stops Update

William Co

2025-08-04

This report documents the geocoding validation process, highlighting discrepancies arising from integrating data sources like Yelp and Yellow Pages. Despite matching entries by phone number, these platforms introduce a high rate of false positives. We apply a variety of filtering to correct for this. We then provide a detailed account of the types of errors observed and offer a numerical estimate of the remaining discrepancies. These errors represent a small proportion of the entire dataset, affecting approximately 17 locations.

Table of contents

1 Progress

1.1 Bug Fixes on Scraping Code

This section details the iterative process of refining our geocoding validation. An initial update addressed a bug in latitude and longitude extraction from the "RVers and Travelers" website. Subsequent analysis focused on resolving discrepancies between different data sources.

1.2 New Matching Process

Initially, we attempted to extract geo-coordinates through a reduction approach. While this method allowed us to observe various geo-coordinates from different sources, it provided no reliable mechanism to determine which coordinates were accurate. This approach proved complex and prone to error due to the scale of the data.

To address these limitations, we developed a more effective strategy based on geographic proximity matching. This method involves calculating the minimum distance between corresponding locations from different data sources, with the closest geographic match considered the most accurate. This distance-based validation approach significantly reduced the number of discrepancies in our dataset. This method operates on the intuition that when two independent data sources yield similar coordinates for the same location, their agreement likely indicates the correct position.

1.3 Previous Matching Process

One big problem with the previous matching process was that we can not accurately rely on input data, as a means to compare. For example, a place might be mark as Zipcode 12345 but its true zipcode is 54321. By using the nearest distance-based validation approach, we bypass this concern.

During this process, we identified postal code input errors as a primary source of location discrepancies. Our initial analysis, revealed a substantial number of locations with discrepancies of less than one mile (see Figure 1).

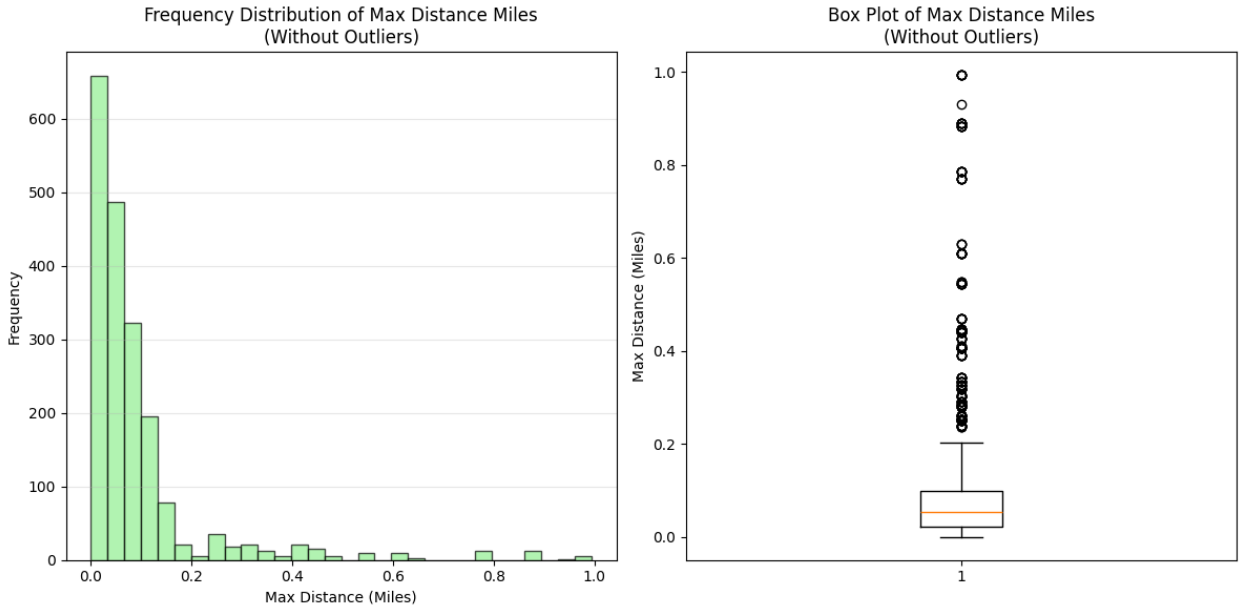


Figure 1: Analysis of discrepancies less than one mile.

Including all outliers, the dataset initially appeared as shown in Figure 2.

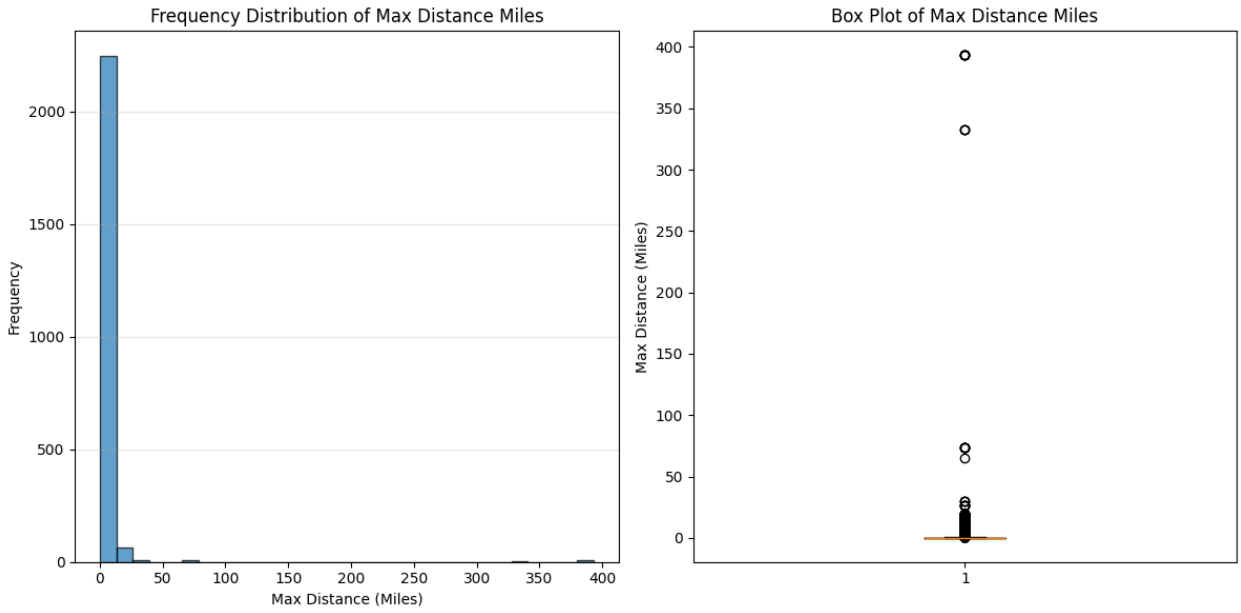


Figure 2: Dataset including all outliers.

After implementing the minimum distance comparison method, we successfully reduced the number of locations with discrepancies greater than one mile from seven to five. The results of this improved approach are visualized in Figure 3.

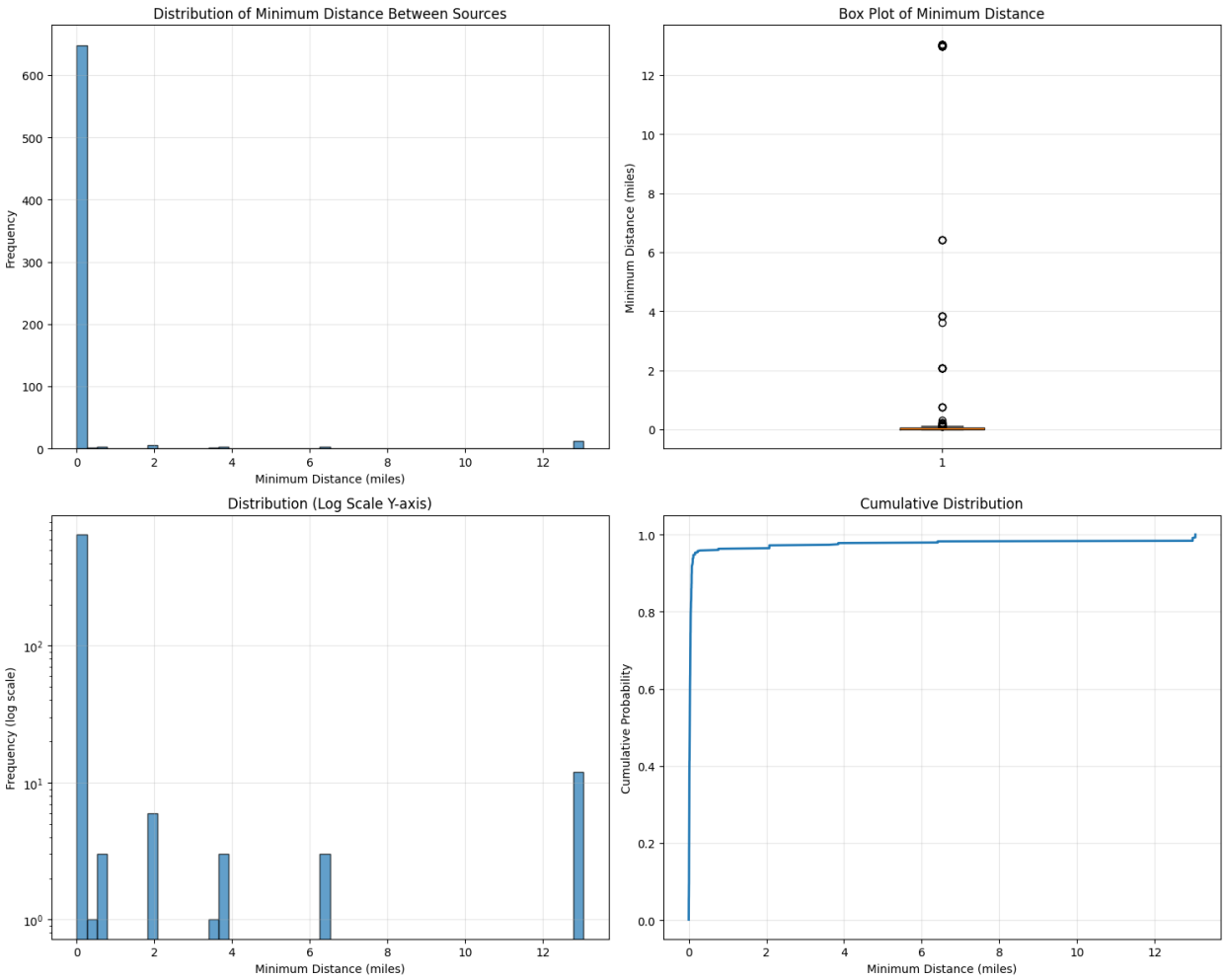


Figure 3: Results of the minimum distance comparison approach.

Our new methodology resulted in only five locations exhibiting discrepancies of more than one mile, representing a significant improvement in overall data accuracy.

2 Manual Verification of Remaining Discrepancies

The five remaining locations with significant discrepancies were manually verified. The analysis revealed no single, discernible pattern to the errors. No single data source was responsible for the errors. Instead, they fall into two main categories:

1. **Geocoding Inaccuracy**: In some cases, the geocoding service returned coordinates with a spatial offset from the actual address. For a given address intended to map to location X, the service returned coordinates at $X + E$, where E represents the spatial error.
2. **Non-Physical Addresses**: Another source of error was the presence of Post Office (P.O.) boxes instead of physical street addresses in the source data.

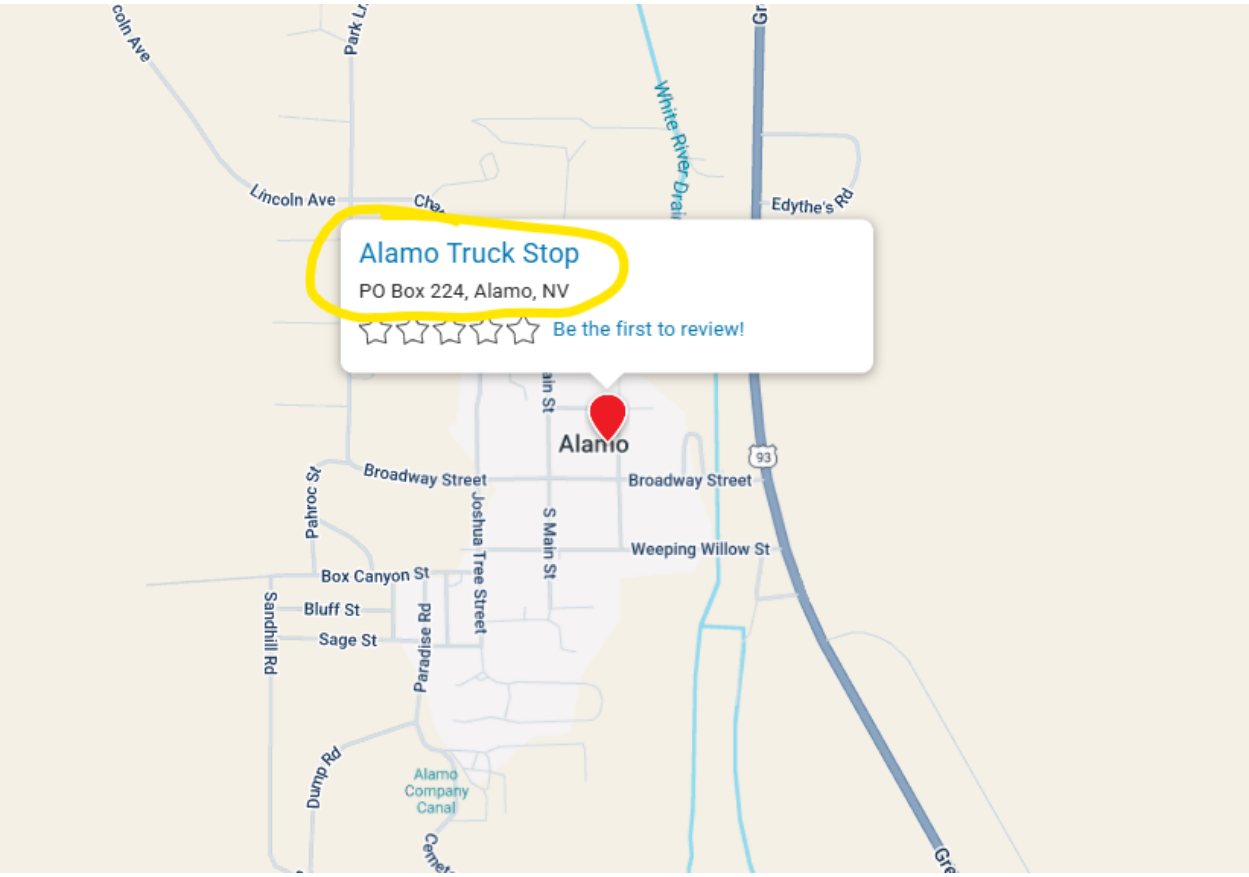


Figure 4: An example of a P.O. Box listed as an address.

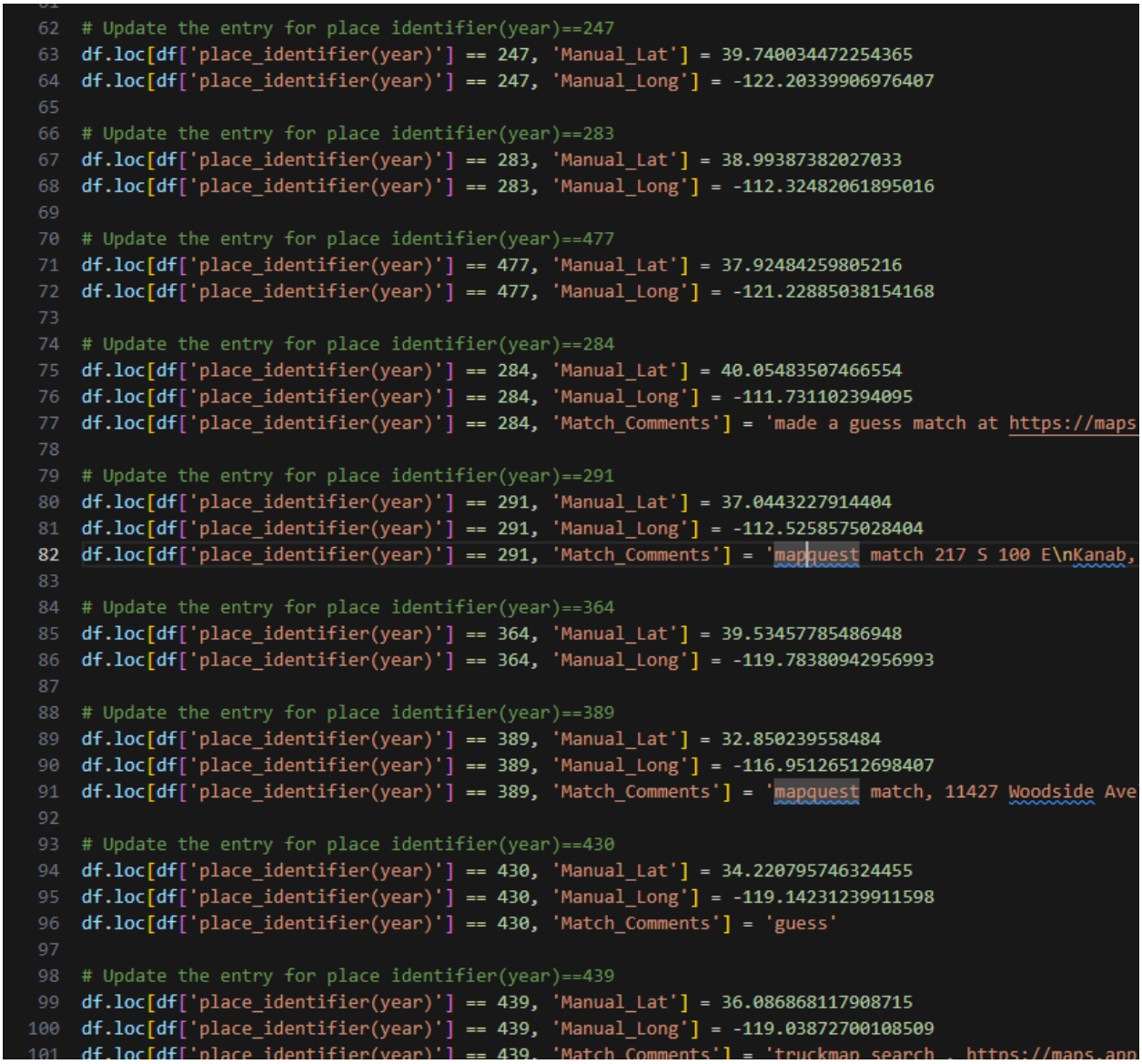
2.1 Results

Following the matching process, all locations have been reconciled to within 200 meters. The next step involves finalizing geo-coordinates by using the midpoints across two slightly different coordinates.

A potential fourth source of geo-coordinates is the Google Geocoding API, which can be used in conjunction with I-exit data. However, this approach is more complex, as not all addresses are formatted correctly and not all entries correspond to highway exits. Not to mention, matching based on exit location can introduce errors greater than one mile.

MapQuest and TruckMap.com are good resources for manually matching these remaining locations.

I completed the manual entries. Below is an example of the process.



```
# Update the entry for place identifier(year)==247
df.loc[df['place_identifier(year)'] == 247, 'Manual_Lat'] = 39.740034472254365
df.loc[df['place_identifier(year)'] == 247, 'Manual_Long'] = -122.20339906976407

# Update the entry for place identifier(year)==283
df.loc[df['place_identifier(year)'] == 283, 'Manual_Lat'] = 38.99387382027033
df.loc[df['place_identifier(year)'] == 283, 'Manual_Long'] = -112.32482061895016

# Update the entry for place identifier(year)==477
df.loc[df['place_identifier(year)'] == 477, 'Manual_Lat'] = 37.92484259805216
df.loc[df['place_identifier(year)'] == 477, 'Manual_Long'] = -121.22885038154168

# Update the entry for place identifier(year)==284
df.loc[df['place_identifier(year)'] == 284, 'Manual_Lat'] = 40.05483507466554
df.loc[df['place_identifier(year)'] == 284, 'Manual_Long'] = -111.731102394095
df.loc[df['place_identifier(year)'] == 284, 'Match_Comments'] = 'made a guess match at https://maps

# Update the entry for place identifier(year)==291
df.loc[df['place_identifier(year)'] == 291, 'Manual_Lat'] = 37.0443227914404
df.loc[df['place_identifier(year)'] == 291, 'Manual_Long'] = -112.5258575028404
df.loc[df['place_identifier(year)'] == 291, 'Match_Comments'] = 'mapquest match 217 S 100 E\nKanab,

# Update the entry for place identifier(year)==364
df.loc[df['place_identifier(year)'] == 364, 'Manual_Lat'] = 39.53457785486948
df.loc[df['place_identifier(year)'] == 364, 'Manual_Long'] = -119.78380942956993

# Update the entry for place identifier(year)==389
df.loc[df['place_identifier(year)'] == 389, 'Manual_Lat'] = 32.850239558484
df.loc[df['place_identifier(year)'] == 389, 'Manual_Long'] = -116.95126512698407
df.loc[df['place_identifier(year)'] == 389, 'Match_Comments'] = 'mapquest match, 11427 Woodside Ave

# Update the entry for place identifier(year)==430
df.loc[df['place_identifier(year)'] == 430, 'Manual_Lat'] = 34.220795746324455
df.loc[df['place_identifier(year)'] == 430, 'Manual_Long'] = -119.14231239911598
df.loc[df['place_identifier(year)'] == 430, 'Match_Comments'] = 'guess'

# Update the entry for place identifier(year)==439
df.loc[df['place_identifier(year)'] == 439, 'Manual_Lat'] = 36.086868117908715
df.loc[df['place_identifier(year)'] == 439, 'Manual_Long'] = -119.03872700108509
df.loc[df['place_identifier(year)'] == 439, 'Match_Comments'] = 'truckmap search , https://maps.app
```

Figure 5: Manual Matches Example

After completing these steps, 13 locations remain that require manual correction. Most of these are old (2008 and earlier) entries, which likely account for the observed errors. No consistent pattern was identified among these remaining discrepancies (aside from age).

The finalized geocoded dataset is available at the following link: Final Geocoded Dataset (10.csv)

Given the small number of unresolved cases, further automation is not justified, nor is it likely to be reliable. Though, we are open hearing any other thoughts on the matter.

these leaves two options. Manual matching against geo-coordinate (Google Geocoding API & I-Exit) matching.