Comparable property finder

2023 Spring Quarter Internship Project

Claire Boyd
ckboyd@uchicago.edu

COOK COUNTY
ASSESSOR'S
OFFICE

Motivating Question

Which properties are most informative to determining the assessed value of any given residential property?

Motivating Question

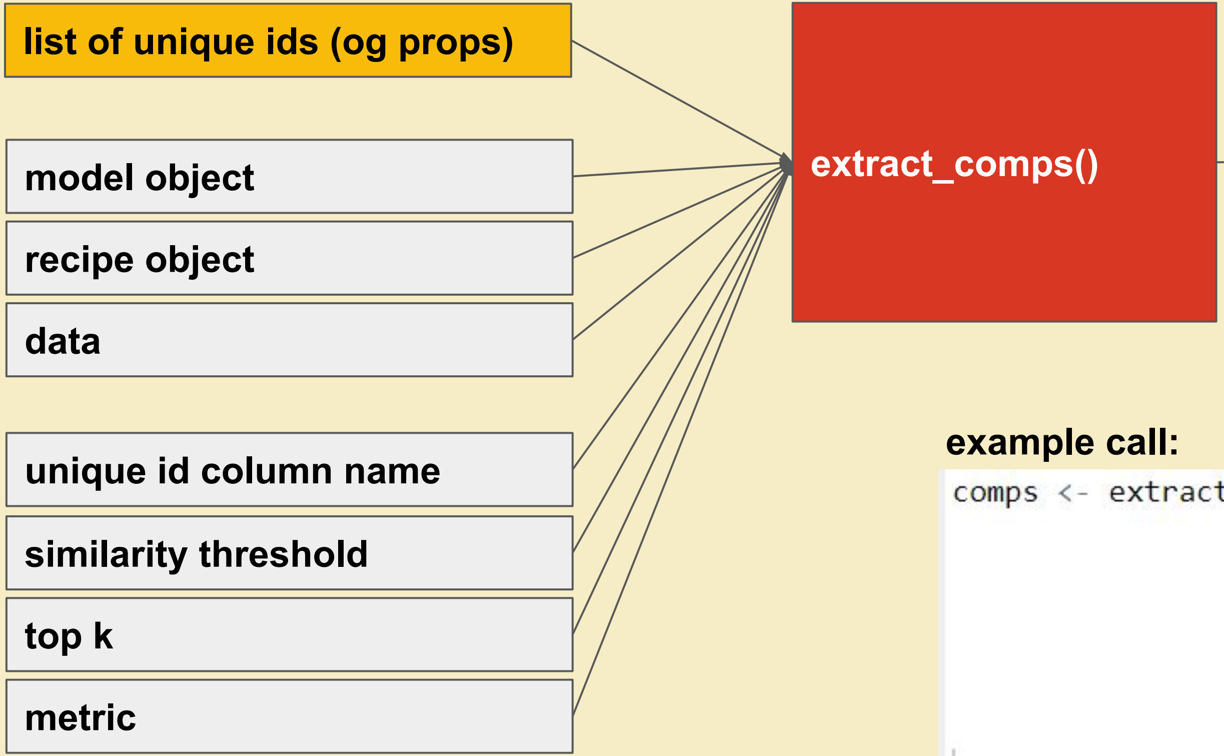
Said differently:

What are the most comparable properties to any given residential property?

How are “comps” currently determined?

- Given the complexity of current model used to assess property value (gradient boosted tree model), it has previously been hard to pull “comps”.
- Right now, “comps” are roughly determined by selecting properties that are in the same neighborhood and have similar values for the top 10 property characteristics, given that these are the most important model features.
- **Problem:** This approach is imprecise, time-consuming to pull manually, and could potentially be inaccurate (making a lot of assumptions). *There could be a better solution!*

Presenting: New & Improved “Comp Finder”



list of unique ids (og props)

model object

recipe object

data

unique id column name

similarity threshold

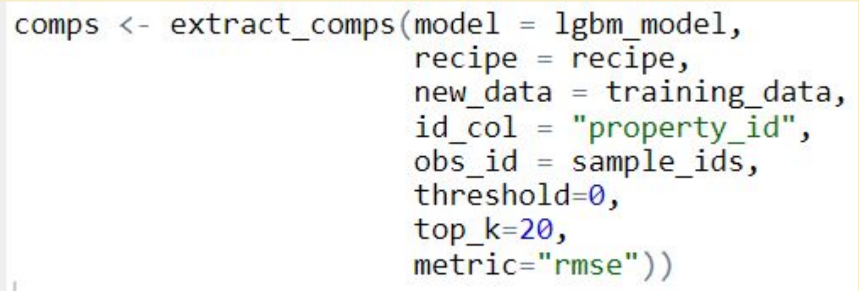
top k

metric

extract_comps()

list of comps for each unique id (og props)

example call:



```
comps <- extract_comps(model = lgbm_model,
                       recipe = recipe,
                       new_data = training_data,
                       id_col = "property_id",
                       obs_id = sample_ids,
                       threshold=0,
                       top_k=20,
                       metric="rmse"))
```

Presenting: New & Improved “Comp Finder”

Decomposition of **extract_comps()**:

1. Get leaf assignments for each tree, using the input data and model object
   a. **Returns:** a large matrix with values for each unique id, for each tree.
2. Get weights for each tree, using a measure of error (e.g. “rmse”)
   a. **Returns:** a vector of weights the size of the number of trees in the model (sums to 1)
3. Calculate similarity scores to original property, using leaf assignments and weights
   a. **Returns**: a datatable with unique id, original property, similarity score to original property
4. Retrieve comps for original properties
   a. **Final result:** a named list of the top k properties by similarity score for each of the original properties provided.

Does it work?

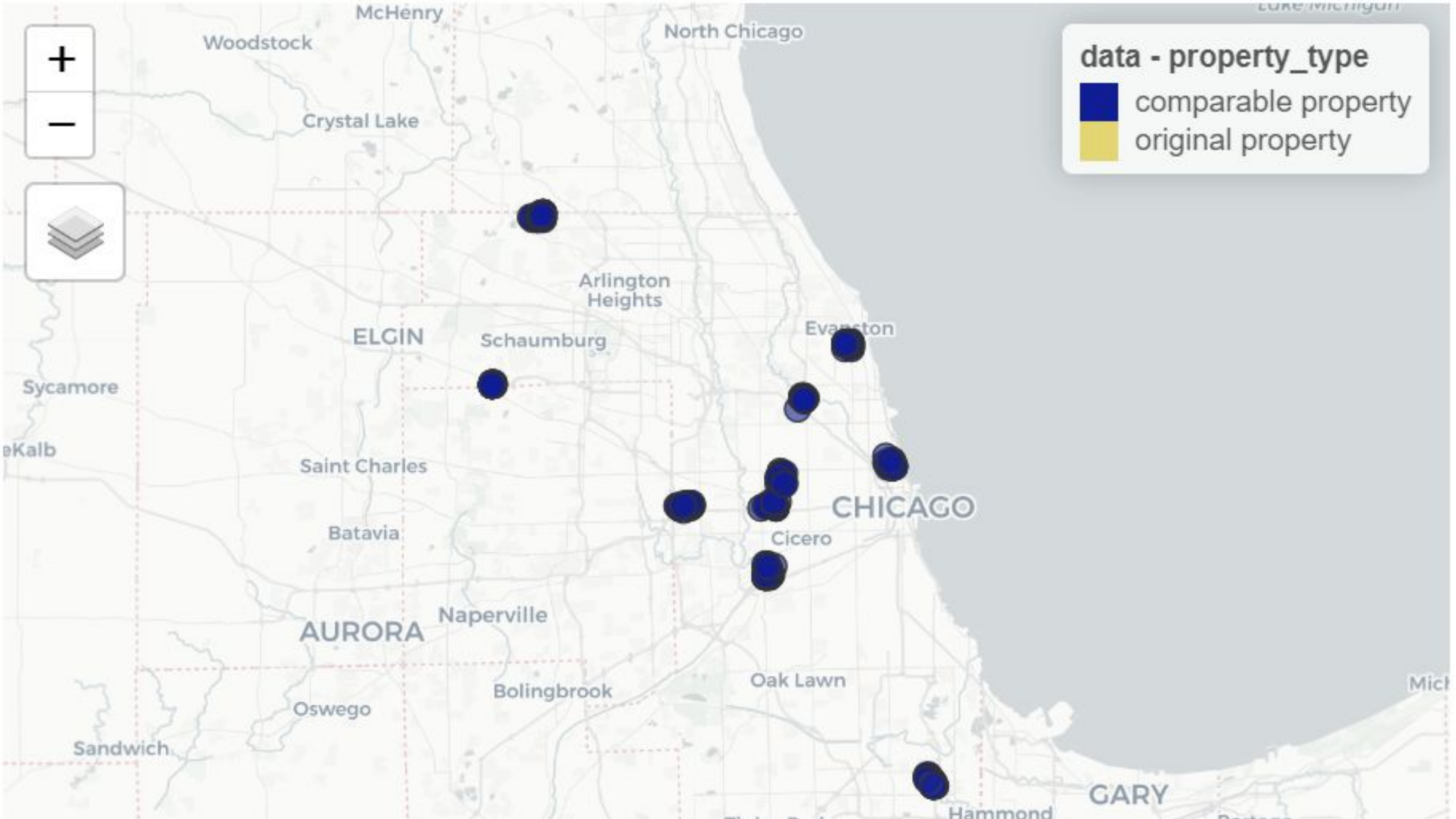
Analysis (for 10 randomly selected properties)

Methodology

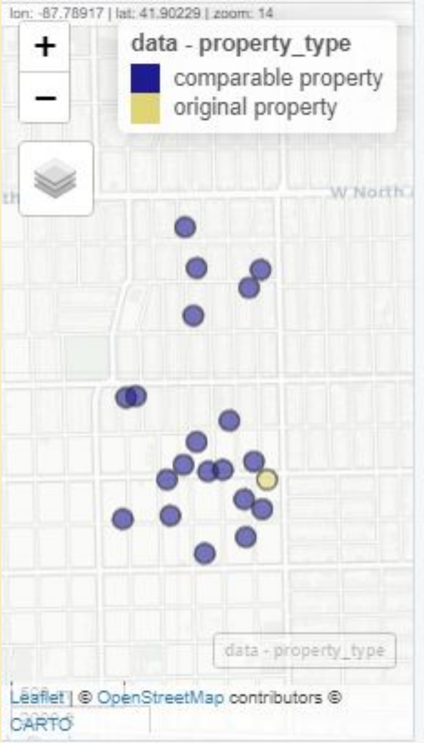
- Sampled 10 randomly drawn unique ids from the training data
- Extracted 20 comps for each randomly drawn unique id

Data

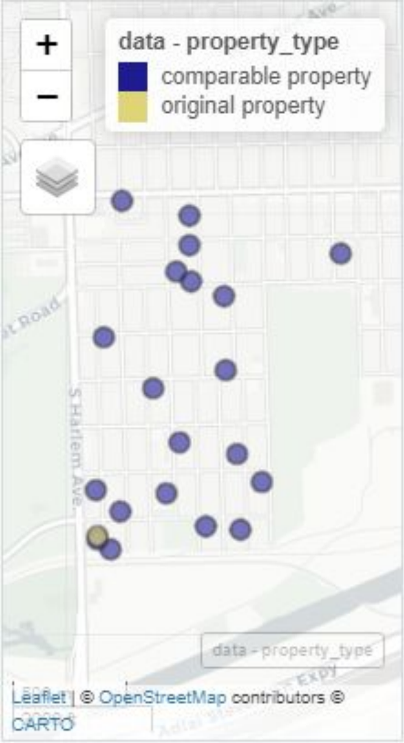
- 210 observations (10 original properties, 20 comps for each)
- Characteristics:
  - All training data features
  - Property ID (manually created unique ID)
  - Base ID, or the unique ID of the original property the comp is related to
  - Similarity Score to original property (NA if it is the original property


+
−
data - property_type
🟦 comparable property
🟨 original property
Lake Michigan
McHenry
Woodstock
North Chicago
Crystal Lake
Arlington Heights
ELGIN
Schaumburg
Evanston
Sycamore
DeKalb
Saint Charles
CHICAGO
Cicero
Batavia
Naperville
AURORA
Oak Lawn
Bolingbrook
Oswego
Sandwich
Mich
GARY
Hammond

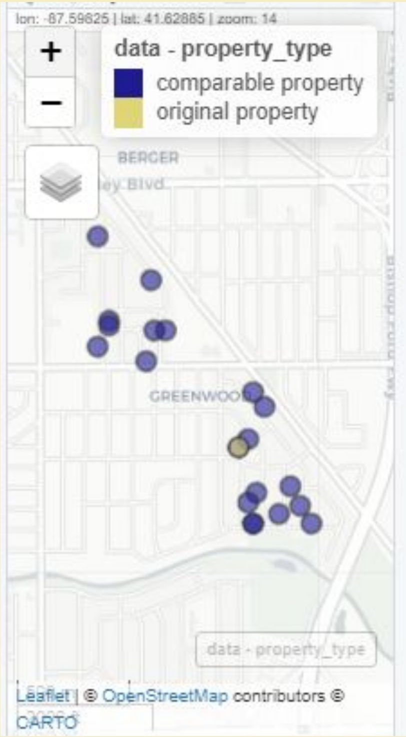
Oak Park


lon: -87.78917 | lat: 41.90229 | zoom: 14
+
−
data - property_type
comparable property
original property
W North
data - property_type
Leaflet | © OpenStreetMap contributors © CARTO

Stickney

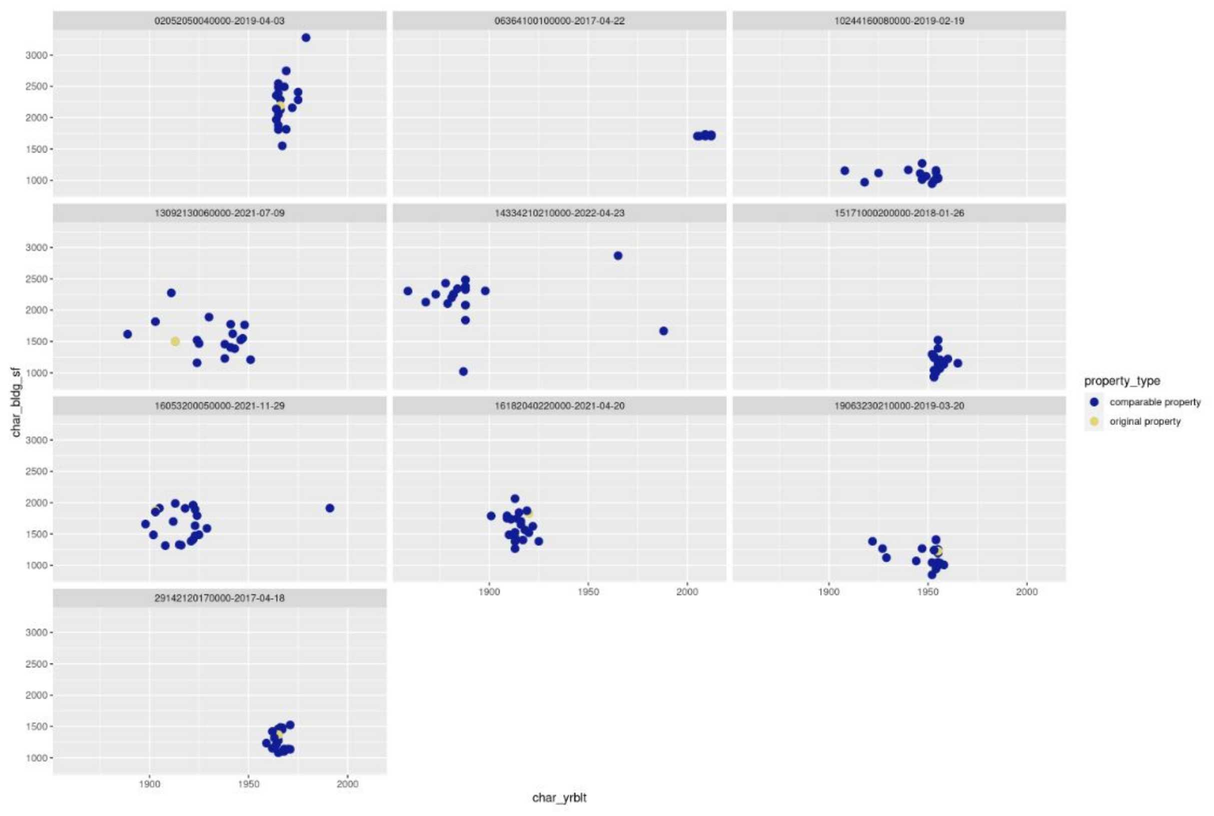


Thornton

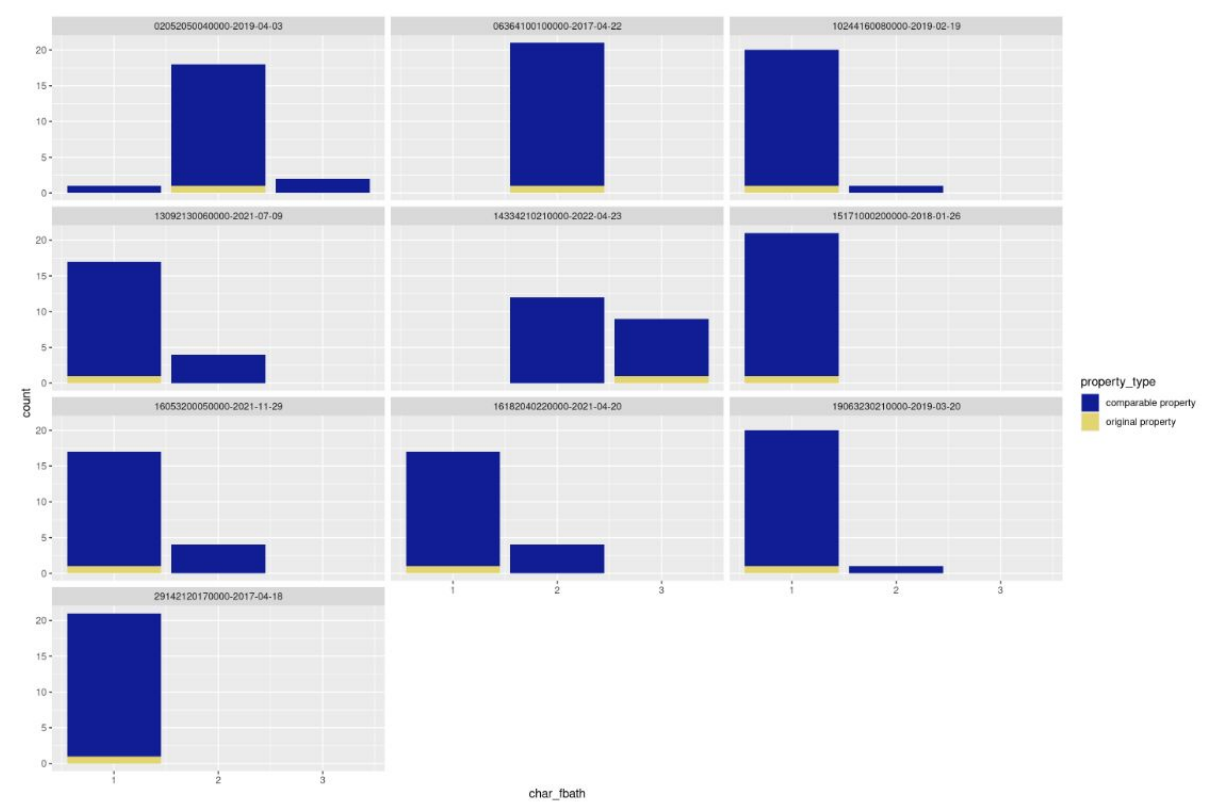


All 20 comps identified for each of the 10 randomly drawn properties **share the same township**, and all but 1 share all of the **same neighborhoods**.

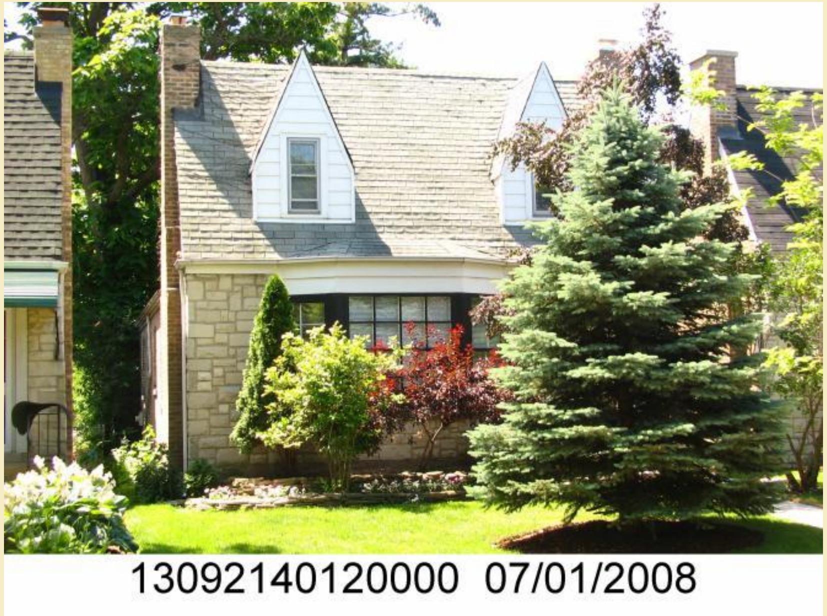
Comps are generally clustered by year built and building square footage.



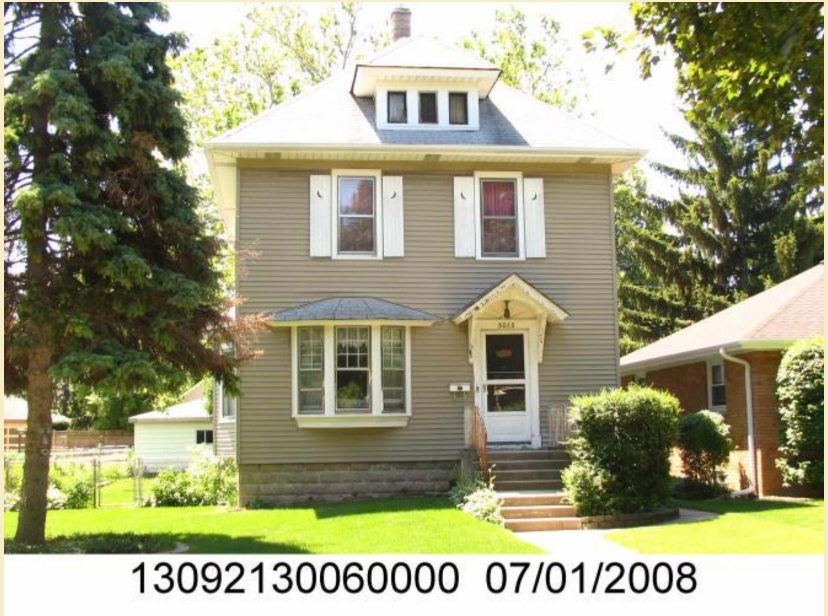
Comps also largely share the number of full baths as the original property.



Original Property



Most Comparable Property

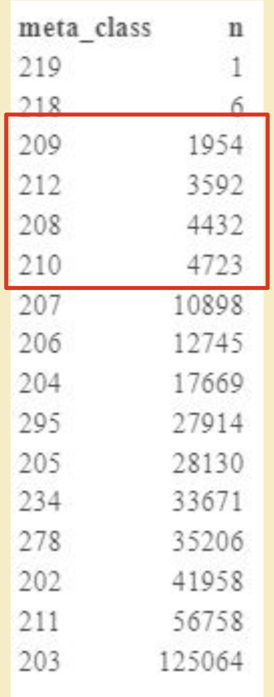


Similarity Score: 0.4694836

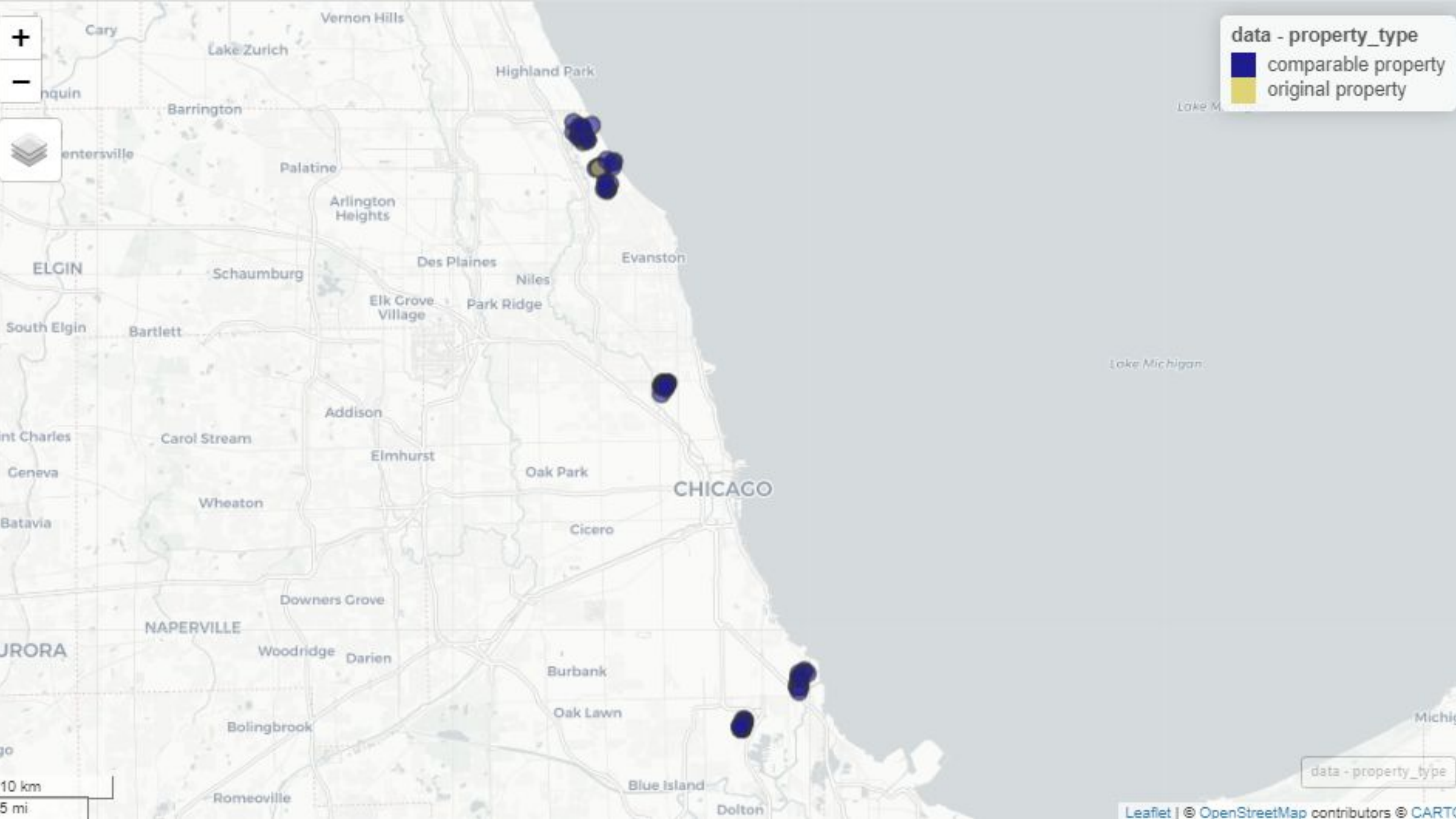
Analysis (for edge case property types)

Methodology

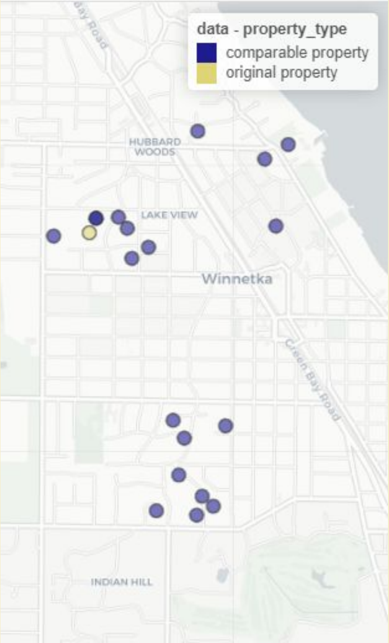
- Identified “rare” property types (209, 212, 208, 210), which have between 1000-5000 observations in the training data.
- Sampled 10 randomly drawn unique ids from each group, as well as 1 random id for a 209 property with a sales_price of over $5M.
- Extracted 20 comps for each randomly drawn unique id



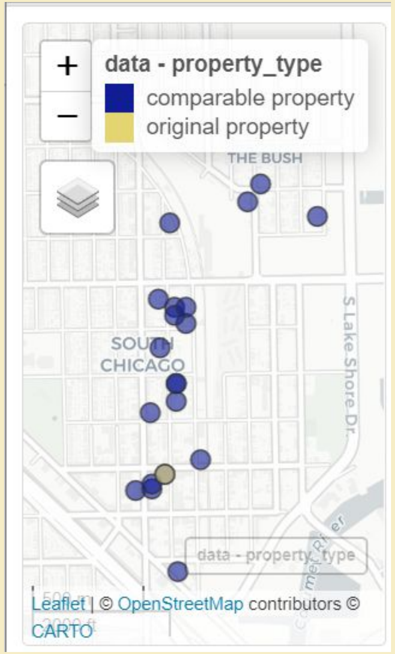
meta_class	n
219	1
218	6
209	1954
212	3592
208	4432
210	4723
207	10898
206	12745
204	17669
295	27914
205	28130
234	33671
278	35206
202	41958
211	56758
203	125064


data - property_type
comparable property
original property
Vernon Hills
Cary
Lake Zurich
Highland Park
nquin
Barrington
entersville
Palatine
Arlington Heights
Evanston
Des Plaines
Niles
ELGIN
Schaumburg
Elk Grove Village
Park Ridge
South Elgin
Bartlett
Lake Michigan
Addison
nt Charles
Carol Stream
Elmhurst
Oak Park
Geneva
CHICAGO
Wheaton
Batavia
Cicero
Downers Grove
NAPERVILLE
URORA
Woodridge
Darien
Burbank
Oak Lawn
Bolingbrook
go
Michi
Blue Island
Romeoville
Dolton
10 km
5 mi
data - property_type
Leaflet | © OpenStreetMap contributors © CARTO

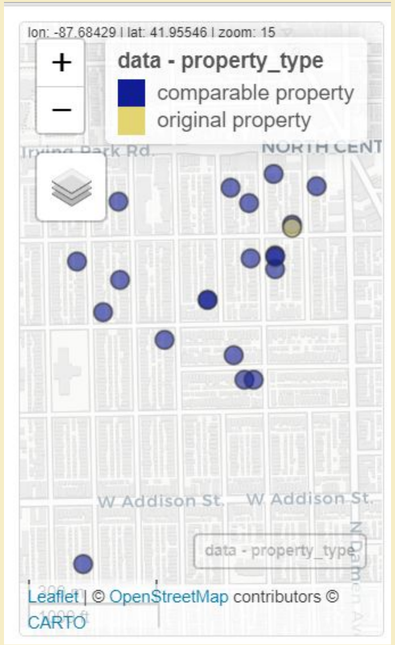
New Trier



Hyde Park

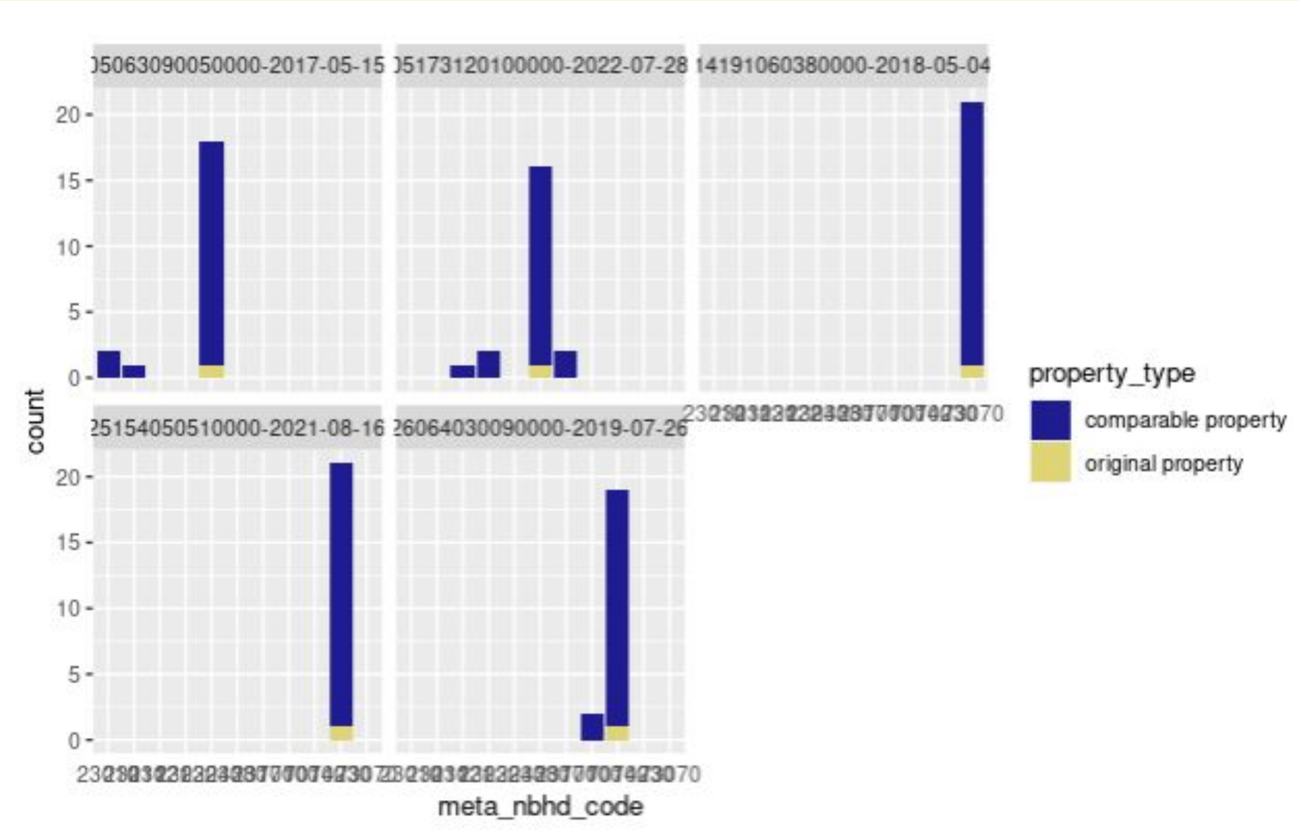


Lake View



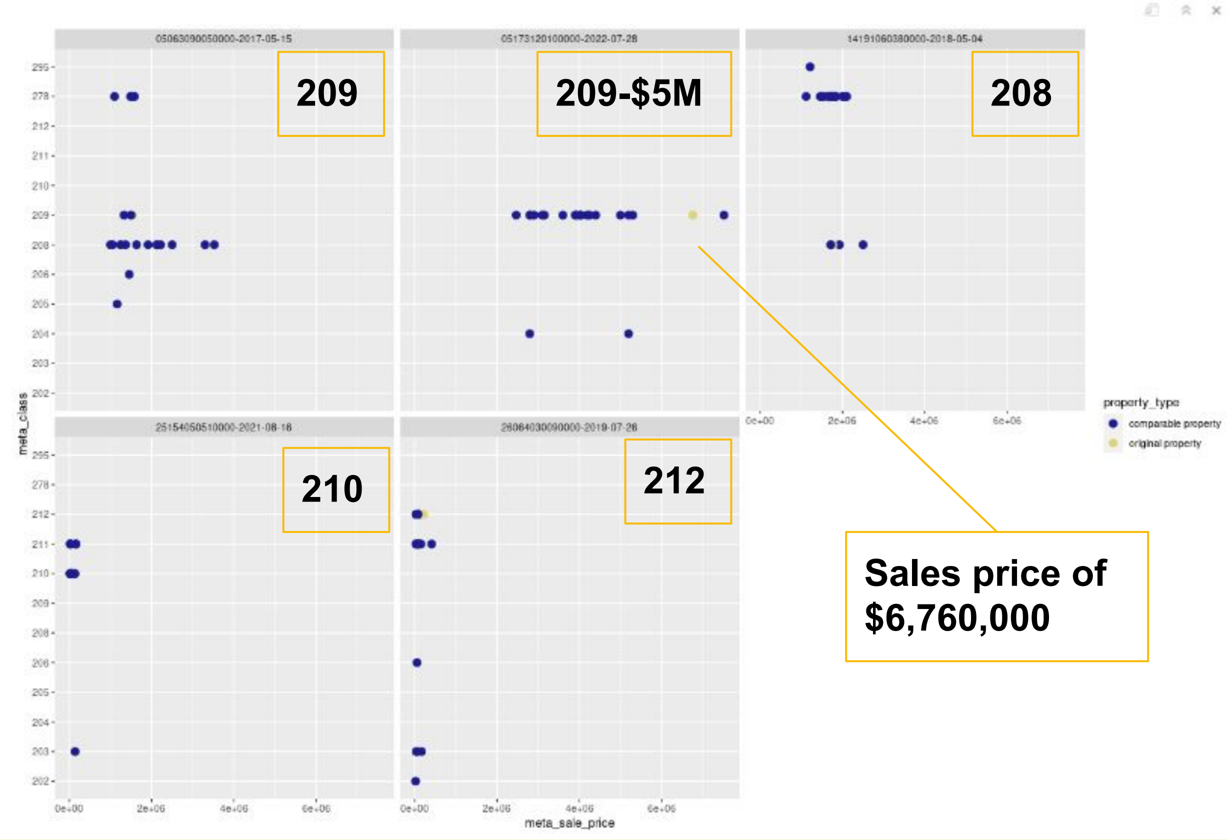
All 20 comps identified for each of the 5 edge case properties, all **share the same township** and secondary school district, and largely share the same elementary school district (with some slight variation).

Unlike the randomly drawn properties, these “rarer” properties vary a bit more in terms of neighborhood code, yet are still clustered around the original property’s neighborhood.



In terms of sales price and property type, comps pulled for these “rarer” properties seem to either align more on the sales price or the property type.

Generally, as sales price increased, there was more clustering at the property type instead of the sales price.



209

209-$5M

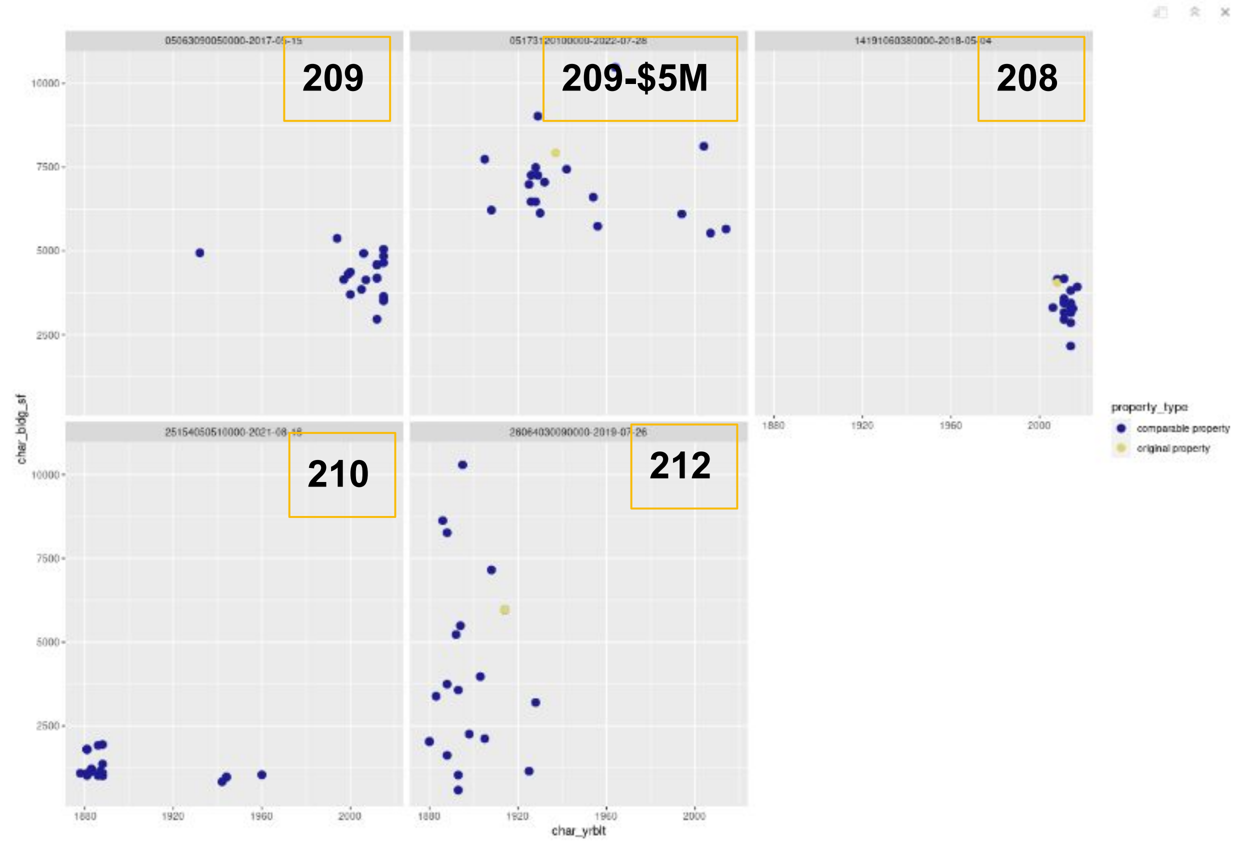
208

210

212

Sales price of $6,760,000

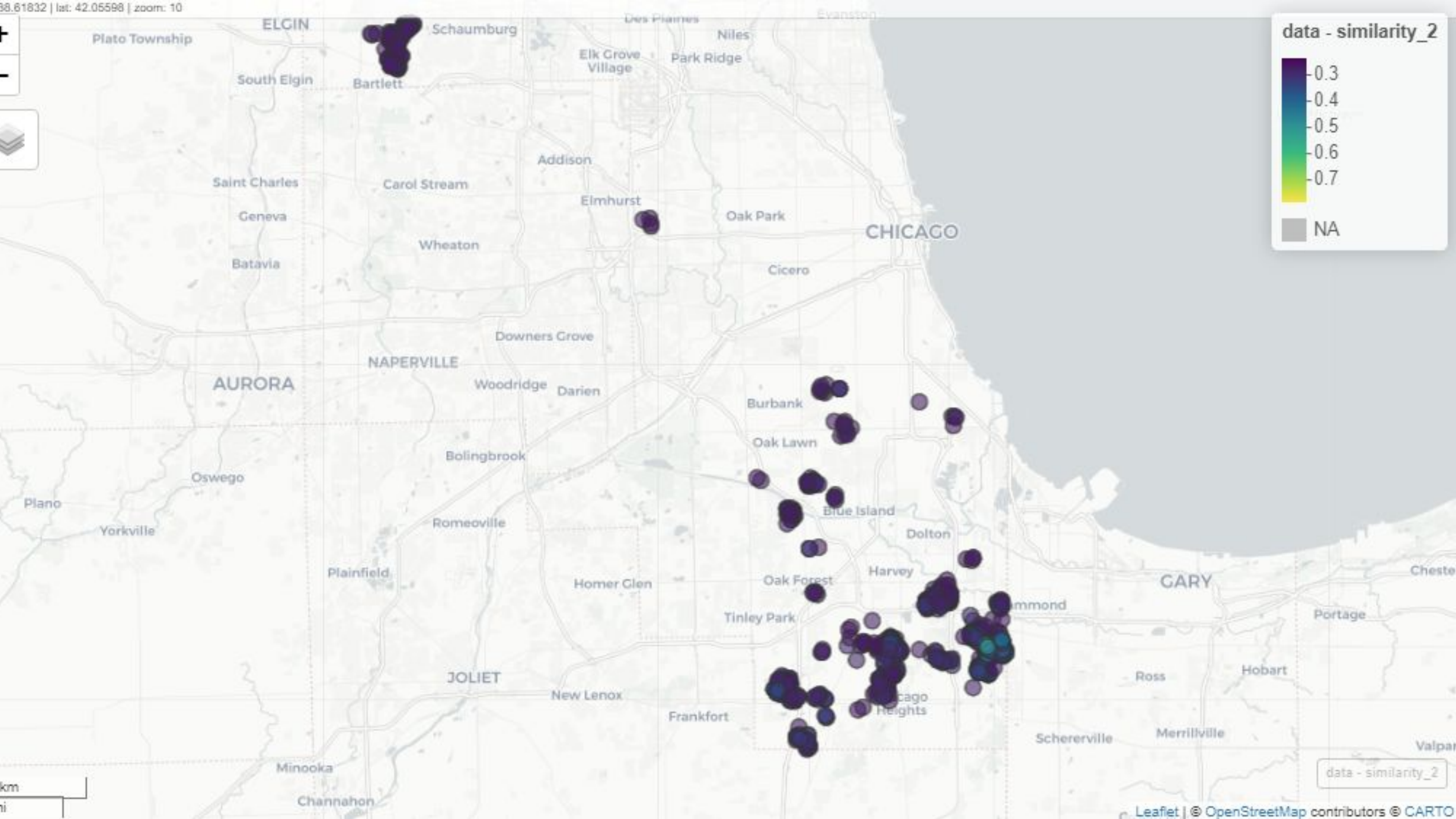
By year built and square footage, the comps for “rarer” property types cluster more consistently for smaller properties (e.g. property types 208, 209 and 210), and are a bit more scattered for original properties with larger square footage (e.g. 209-$5M, 212).

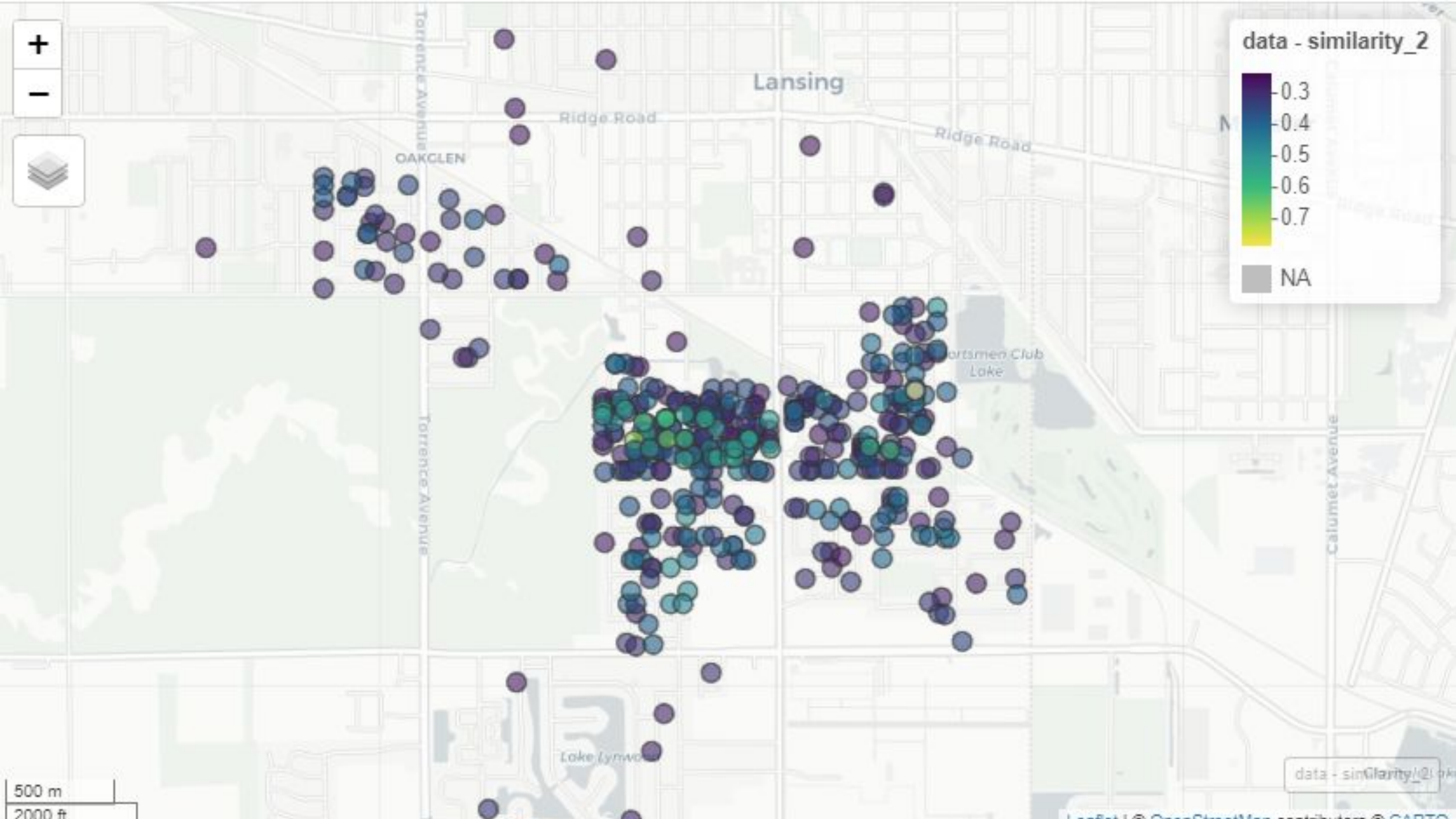


Analysis (for 1000 comps for a property)

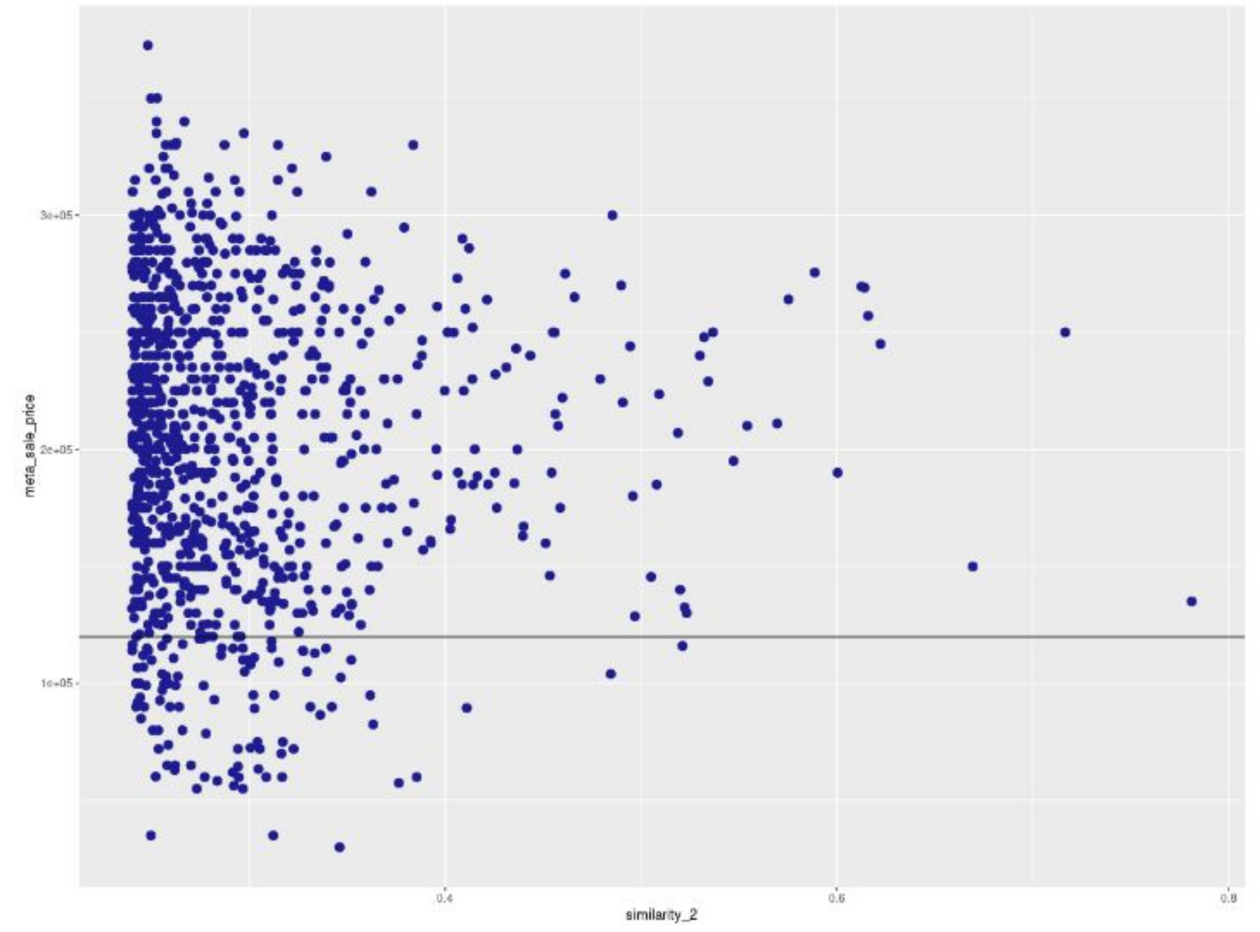
Methodology

- Identified one property at random
- Extracted 1000 comps for that one property to compare similarity score with specific characteristics.

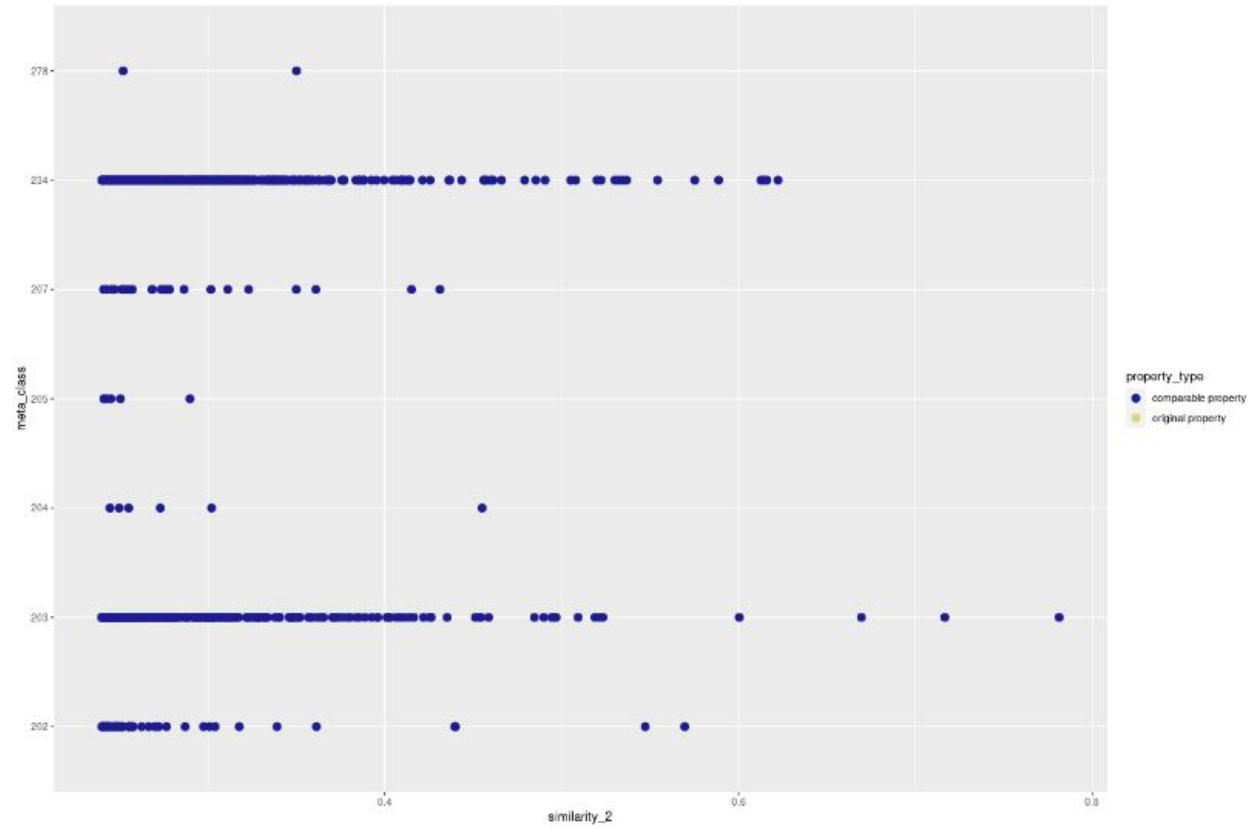

38.61832 | lat: 42.05598 | zoom: 10
+
−
data - similarity_2
0.3
0.4
0.5
0.6
0.7
NA
ELGIN
Schaumburg
Des Plaines
Evanston
Niles
Plato Township
South Elgin
Bartlett
Elk Grove Village
Park Ridge
Addison
Saint Charles
Carol Stream
Elmhurst
Oak Park
Geneva
CHICAGO
Wheaton
Batavia
Cicero
Downers Grove
NAPERVILLE
AURORA
Woodridge
Darien
Burbank
Oak Lawn
Bolingbrook
Oswego
Plano
Yorkville
Romeoville
Blue Island
Dolton
Plainfield
Homer Glen
Oak Forest
Harvey
GARY
Hammond
Tinley Park
Portage
Chesterton
JOLIET
New Lenox
Frankfort
Chicago Heights
Ross
Hobart
Schererville
Merrillville
Valparaiso
Minooka
Channahon
km
mi
data - similarity_2
Leaflet | © OpenStreetMap contributors © CARTO


+
−
data - similarity_2
0.3
0.4
0.5
0.6
0.7
NA
Lansing
Ridge Road
Ridge Road
OAKGLEN
Torrence Avenue
Torrence Avenue
Calumet Avenue
ortsmen Club Lake
Lake Lynwo
500 m
2000 ft
data - similarity_2
Leaflet | © OpenStreetMap contributors © CARTO

Looking at the similarity score compared to sale price, there is a lot of variation when the similarity score is low, but a little more consistency for the higher similarity scores (though not much).

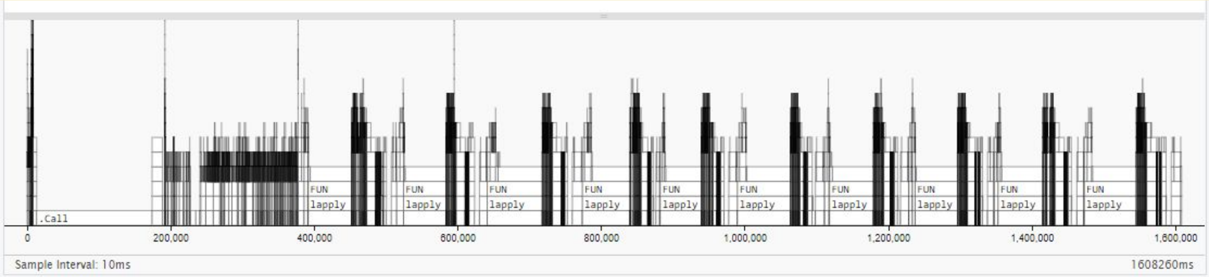


There is also pretty significant variation in terms of property type, but we learned from the “rarer” property type analysis, that this variation is true even within the top 20 comps.



Core Challenges

1. Comps extracted are only as good as the current model itself
   a. Any bias or overestimation of the model is translated to the comparable properties identified
2. Computational efficiency
   a. Due to design of **extract_comps()**, the original property leaf assignments need to be compared to every other property's leaf assignments which an expensive operation computationally.
   b. We have made some improvements to speed this up, but more can be done to streamline how comps can be extracted in parallel for multiple ids (see flame graph below for 10 ids)


.Call
FUN
lapply
FUN
lapply
FUN
lapply
FUN
lapply
FUN
lapply
FUN
lapply
FUN
lapply
FUN
lapply
FUN
lapply
FUN
lapply
Sample Interval: 10ms
1608260ms