BinoML: Supervised Ranking for Automatic Building Labeling

Puyang Ma
puyanma@amazon.com
Amazon
Bellevue, WA, USA

Ravi Garg
ravigarg@amazon.com
Amazon
Bellevue, WA, USA

Mohamed Moustafa
mmoustm@amazon.com
Amazon
Bellevue, WA, USA

ABSTRACT

Building numbers shown on building outlines of a map are important information for guiding delivery associates to the correct building of a package's recipient. Intuitively, the more labeled buildings are present in our map, the less likely to misplace an order in addition to other benefits such as delivery efficiency as drivers get better visual cues about building positions. Although there are free and collaborative projects for creating geographic database of the world, such as the OpenStreetMap (OSM) [2] which also supplies building outlines along with their building numbers, many building outlines still remain unlabeled in many U.S. regions and other countries. Hence, we are interested in developing models that can automatically add building numbers with ≥ 99% precision to unlabeled buildings across geographies with low to medium building number coverage. In this paper, we describe a ML model which in offline results showed 2% to 12% increase in building number coverage in some US regions compared to that of the OSM. The proposed model can also be applied to improve the building number coverage of other countries after fine-tuning to those new regions.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Information systems → Geographic information systems**.

KEYWORDS

Geospatial Data, Machine Learning, GPS Points, Map Features

ACM Reference Format:
Puyang Ma, Ravi Garg, and Mohamed Moustafa. 2022. BinoML: Supervised Ranking for Automatic Building Labeling. In *The 1st ACM SIGSPATIAL International Workshop on Spatial Big Data and AI for Industrial Applications (GeoIndustry '22), November 1, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3557922.3567481

1 INTRODUCTION

Some building footprints on a map carry labels such as building numbers, which are very helpful information for delivery associates to quickly find correct drop-off locations. Figure 1 shows some examples of labeled building outlines from the OSM. There is also an abundance of unlabeled buildings as shown in Figure 2. As part of our efforts in Last Mile to improve delivery efficiency and reduce delivered but not received (DNR) events, previously there have been several attempts to address the issue of low to medium building number (BN) coverage (≤ 70%) in OSM of the US and other countries. The previous best approach is a scalable heuristic algorithm that matches addresses to building outlines and use the building number in an address text to label its matched building. The inputs to the heuristic method include a Geocode1 file which contains latitude, longitude point (a geocode) representation of each address, and building polygons from OSM. The method assigns an address to its nearest building, and calculates a confidence score which indicates how confident an assignment is. To achieve a production level precision of ≥ 99% precision, matches with less than 0.95 confidence and ambiguous numbers for a building are removed. The heuristic method is as follows.

(1) For each address:
   (a) Get its latitude, longitude representation (geocode) from the Geocode1 file.
   (b) Among building candidates in 30m radius, choose the nearest building.
   (c) Compute a confidence score based on the geocode distance ($dist_i$) to the chosen building and to all other candidates. $Confidence = \alpha^{min(dist_i)} * \frac{max(\beta^{dist_i})}{\sum \beta^{dist_i}}$, where α and β are fixed values less than 1.
(2) Remove matches that label a building with ambiguous numbers.
(3) Remove matches with confidence score less than 0.95.

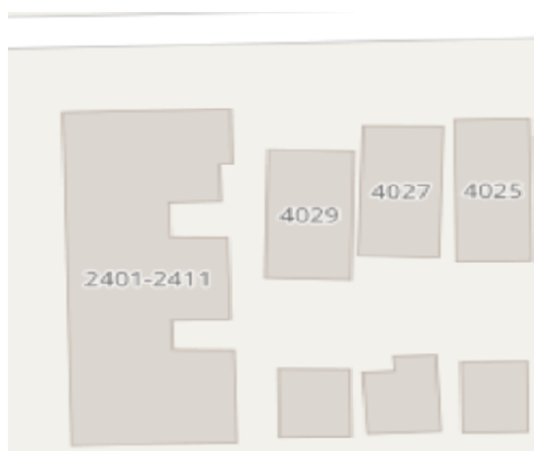

Figure 1: OSM building outlines with their building numbers. Map data ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

By analyzing the unmatched addresses of the heurisitc, we saw opportunities in developing a machine learning model to further

GeoIndustry '22, November 1, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9535-9/22/11.
https://doi.org/10.1145/3557922.3567481

Figure 2: Unlabeled building outlines. Map data ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

increase the BN coverage across different countries. From a sample of matching results of the heuristic method for a region, we observed two major types of unmatched addresses: About 23% are due to low confidence values, and about 31% are due to ambiguous building numbers (a building matched with multiple addresses of different building numbers has ambiguous number). The remaining cases are due to missing building outlines from the map. To reduce the number of these two major types thereby increasing the number of labeled buildings, we developed a pair-wise ranking model (BinoML) for automatic building labeling with high precision. The primary contributions of our paper are:

- We translated this assigning addresses to buildings problem into a supervised ML ranking problem, which uses features such as building areas, past delivery scans around a building, sequence of addresses and more to label 2 to 5 times more unlabeled buildings than the heuristic does.
- We created a method to automatically assign positional orders of buildings along a road for both simple and relatively complex regions. These positional orders enable the use of neighbor building information to determine if a building is a reasonable choice for an address.
- We demonstrated that offline results of BinoML have shown further building number coverage improvements even for high coverage (> 73%) US regions. With model fine-tuning to a new country, BinoML also has the potential to significantly improve BN coverage for currently low coverage countries.

2 RELATED WORK

2.1 Bipartite graph

The matching problem can be thought of as a bipartite graph [3] between buildings and addresses as shown in Figure 3. The rectangles represent buildings, and circles with numbers represent addresses. An edge exists between a building and an address if that building is a candidate for that address. If we construct this bipartite graph for matching, we may consider using **maximum flow algorithms** [1] where there are a sink (circle t) and a source (circle s). Since each address can only be assigned to a building once, the edge capacity between a building and an address is 1, and we can set the edge capacity between an address and the sink as 1 to ensure assigning an address to just one building. However, we do not know if a building is single-numbered or multi-numbered. To finish setting up the maximum flow problem we would need to know how many unique addresses should be matched to a particular building. Another deficiency of such bipartite graph methods is time consuming computation, thus posing scalability issues.

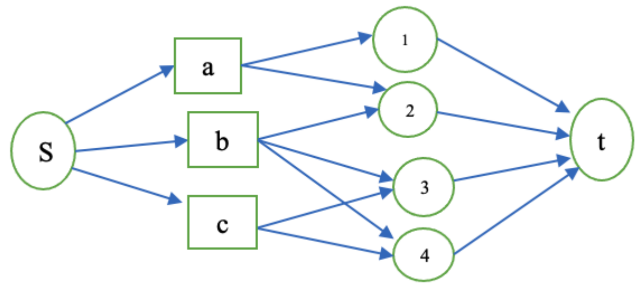

Figure 3: Imagined bipartite graph between buildings (a, b, c) and addresses (1,2,3,4).

2.2 Ranking

Since a building may have multiple building numbers or a single building number, a natural way to frame this matching problem is to choose the best building among building candidates for an address. Previously, there was a delivery point (DP) model [4] that uses ranking to choose the best delivery scan point as the geocode (latitude, longitude representation) of an address, where a scan point is a GPS point generated when a driver scans a package for delivery completion. Although drivers are asked to scan a package only at its dropping location to indicate successful delivery, in reality there is no guarantee that drivers do not scan packages in the car, or across the street, or somewhere else for convenience. Even if all drivers strictly follow the procedure, their package scan points can be inaccurate in urban environments. Thus the author of the DP model [4] used a pair-wise ranking model to compare delivery scan points for an address and choose the best point for package dropping. Figure 4 shows a good delivery point output from the DP model for an address, which is close to the doorstep of the correct building. However, as shown in Figure 5 there are also many cases where the model's delivery point output lies between buildings, in which case it is hard to tell which building the address should be assigned to. We adopted this ranking approach for our problem of assigning an address to its correct building, but we created a new set of building related features and ranked candidate buildings rather than scan points for an address. In this way, an address will be assigned to only one building while a building can be assigned multiple addresses.

3 PROPOSED APPROACH

We built a pairwise ranking model to choose the best building for each address, so that each address will be assigned to just one

Figure 4: Delivery point (red) and package scans (blue) for an address (background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2]).

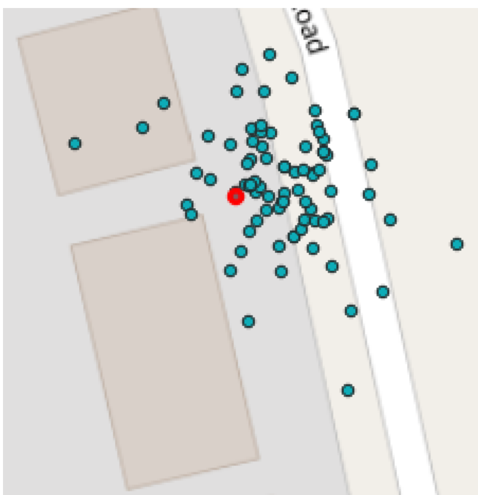

Figure 5: Delivery point (red) and package scans (blue) for an address (background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2]).

building, but a building may have multiple addresses. The input data of the model are 18 months package scans data of a delivery region, a road segment map, and OSM's building outlines within the delivery region's boundary. We pre-processed the data before giving as input to our ML model and these steps are described below.

3.1 Pre-processing

Remove ambiguous addresses with a balking classifier: Some addresses have scans covering a wide range of buildings, and some are garden community addresses that should not be assigned to a single building. Figure 6 shows an example where multiple buildings share the same address text, and for such an address the delivery scans are dispersed among the buildings, so it does not follow the assumption of belonging to only one building. We use an existing balking classifier from the DP model [4] to separate single building addresses (including high-rise buildings) from these kinds of garden community addresses, so that BinoML will balk at garden communities where multiple buildings share the same address.

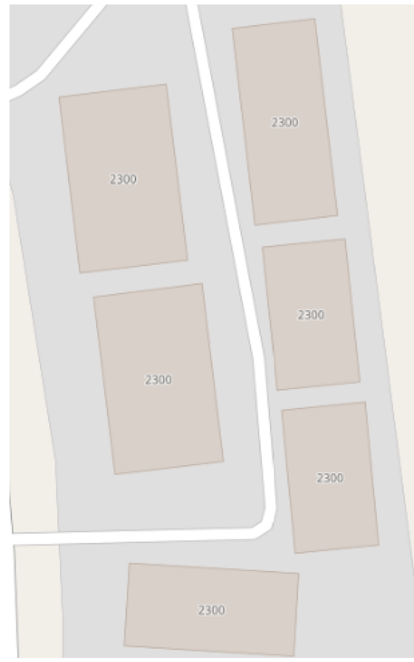

Figure 6: A community where multiple buildings share the same address. Map data ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

Candidate buildings for an address: Since BinoML is a ranking model that selects the best building outline for an address, we need to get candidate buildings for each address. Before getting candidate buildings, we filter out buildings as shown in Figure 7 that do not deserve labels such as attached garages, sheds, etc. We currently do it by removing building outlines of size below 30 squared meters. For each address, we form an envelope from its DP and its maximum KDE score (2D kernel density estimate of scans) point. We expand the envelope by 60 meters vertically and horizontally with a 3969 squared meters area constraint (about the area of a 35m radius circle) to get a final envelope. Buildings that intersect with the final envelope are candidate buildings for the address.

Address normalization: Each input address is in a normalized format, with a structure like "APT Number!Building Number!Street Name!City!County!State!Country". Despite being a normalized format, due to different input habits ad accidental misspellings, the input address may still have different strings for the same word such as "Apt" and "appartment", "room" and "rm". Moreover, presence or absence of spaces can result in different strings for the same address, e.g. "Neal Crest" and "NealCrest" are different strings but actually are the same. We resolve these issues by further normalizing each address using common US address abbreviations and removing unnecessary spaces.

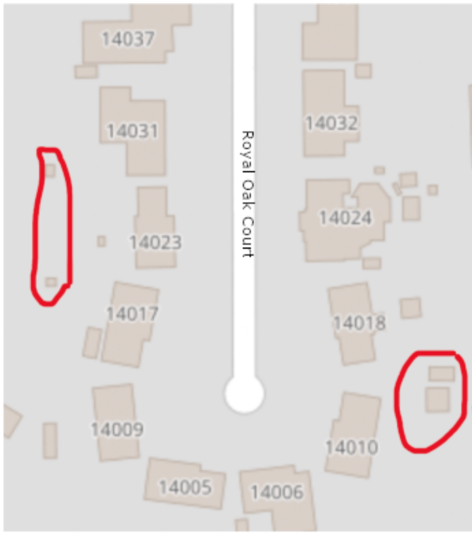

Figure 7: Garage-like buildings in red frames. Background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

Building orders along a road segment: For many cases, building numbers follow sequential order along a street. We capture this information by assigning positional orders to buildings along a road segment, thereby obtaining possible information of a candidate building's neighboring buildings for some feature generation. Automatically assigning building orders can be challenging as it is not given which road segment a building belongs to, and secondly many garage-like buildings tend to be near main buildings and can introduce noise in the building orders. For building and road segment association, we first search for segments within 80m of a building and choose the segment with matching road name as that of the building, and if either the road name or the building's street label is absent we choose the segment closest to the building's centroid. We have also devised a heuristics method to automatically estimate building orders for both simple and relatively complex regions: After associating buildings with a road segment as described before, we shifted the segment to the direction of its associated buildings by the avg distance to their centroids. If the shifted line crosses most buildings, we generate position orders for all of these buildings by projecting their centroids onto the road and assigning sequential numbers based on the projected points' position on the road. Figure 8 illustrates positional orders for buildings of a simple region.

In complex regions, garages or sheds often scatter around main buildings. Figure 9 shows such a region from OSM. After checking other map sources and satellite images, we confirmed that all of those unnumbered outlines in Figure 9 are non-residential buildings like garages or sheds. In this type of regions, we observed that the shifted road segment often does not cross at least 3 of its associated main buildings. To avoid assigning orders to non-residential buildings, we generated convex hulls (Figure 9) of a road's major buildings' (based on relative area) centroids, and only assigned positional orders for buildings intersected with the convex hulls. Figure 10 shows positional orders for buildings of a complex region. A segment may have two sides of buildings as is the case for segment "D" in Figure 10 and positional orders are formed independently on each side. If some buildings on one side of a segment have the same projected point, then they will have the same positional order along that segment, as is the case for the multiple 1s on either side of segment "D".

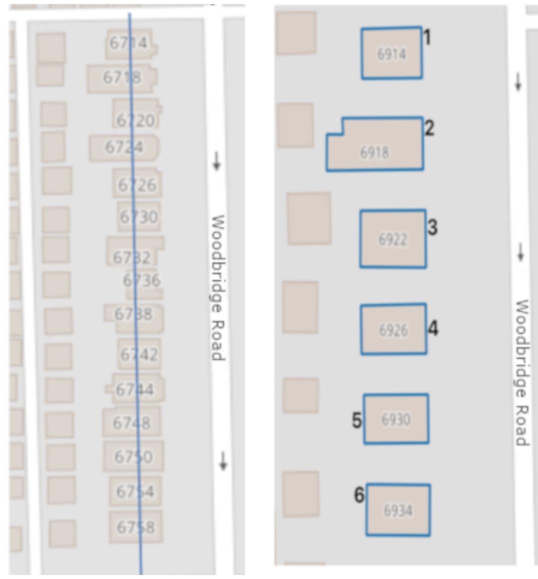

Figure 8: Shifted road segment (blue line on the left) crosses all the segment's associated buildings, and positional orders (bold number) on the right (background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2]).

3.2 Feature vector

There are two types of features used in the ranking model. The first type is building features of a candidate building for an address, and the second type is background features specific to an address but have the same values for all candidate buildings. For the pairwise ranking model, differences between a pair of buildings and some address background features form a feature vector with a binary target of whether the left building is better than the right building for an address. The building features include:

- KDE distance: The shortest distance in meters between a building and the max KDE score point calculated from the most recent 500 scans (or all scans, if fewer than 500) of an address. We use an exponential kernel with a bandwidth of 25m to calculate KDE scores.
- Geocode distance: The shortest distance between a building and the latest DP point of an address.
- In between: Whether the address text has a building number that fits in the candidate building's previous and next building neighbors. The possible values are True, False, and None (in case the previous and next neighbor buildings' numbers are unknown). E.g., for an address with building number 10, if a candidate building's previous neighbor is labeled 9 and the next neighbor is labeled 12 then the value is True.

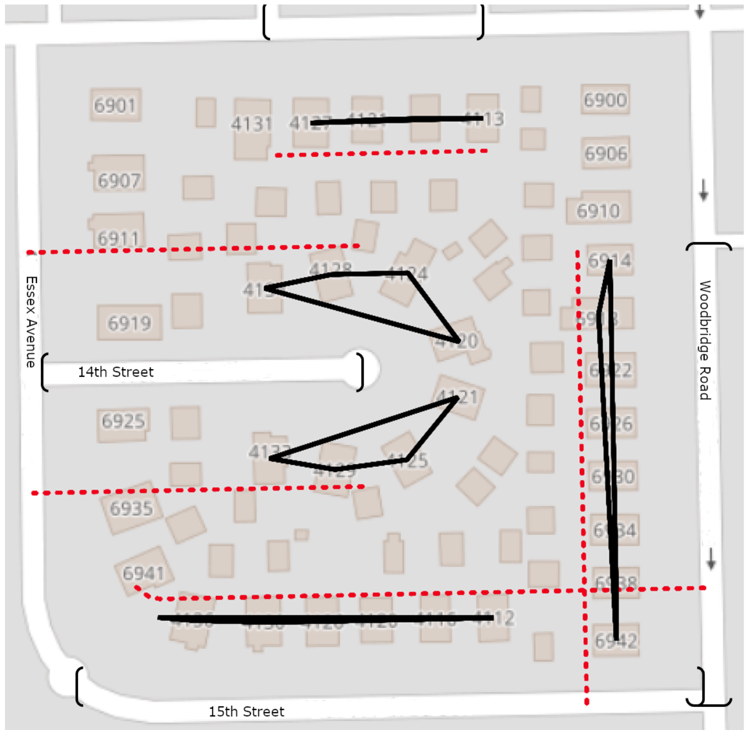

Figure 9: Road segments (in brackets), shifted segments (dashed lines), and convex hulls of building centroids (solid black). Background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

Figure 10: Road segments (left) and positional orders of buildings associated with each segment (right). The capitalized letters indicate association between buildings and road segments. Background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

- Inside building scan share: Ratio of scans in this building among scans inside any building.
- Soft vote share: Each package scan of an address cast a partial vote to a candidate building that is inversely proportional to the distance between the scan and the building. The soft vote share is the fraction of all partial votes a building gets for an address.
- Average scans distances to a building: The average of an address's package scans distances to a building.
- Relative building area: Z-score value of a building outline's area among the areas of all candidate buildings for an address.
- Name difference: The minimum absolute difference between an address's building number and a building's labeled numbers if any exist. For an address with building number 944, the building name difference is 2 compared to a building labeled 946, etc.
- Position mean: The absolute mean of **non-NaN** differences between an address's building number and a building's neighbors' labeled numbers. As shown in Figure 11, for an address with 944 as its building number, consider the candidate building labeled 942: The previous building is labeled 944 and its next building labeled 938, then the position mean is $|\frac{944-944+944-938}{2}| = 3$.

The background features of an address include such information as maximum soft vote share, number of candidate buildings, ratio of scans within 5m and within 20m of a building, and the average building area of candidate buildings. After forming all possible pairs of buildings from candidate buildings of an address, a feature vector is in the form of $(\mathbf{u}-\mathbf{v}, \mathbf{c})$, where $\mathbf{u}$ refers to the left building features, $\mathbf{v}$ refers to the right building features, and $\mathbf{c}$ represents the common background features of an address.

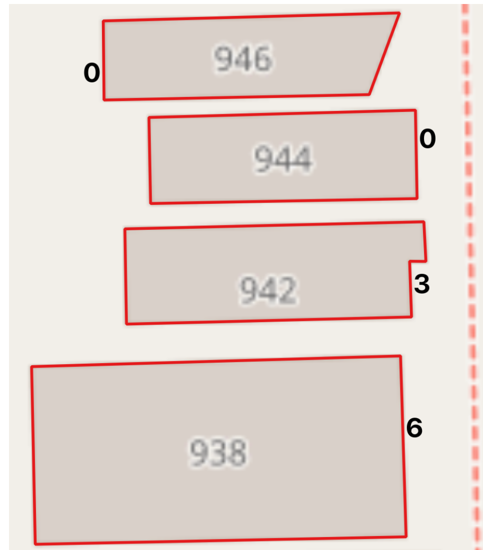

Figure 11: For an address text with building number 944, bold numbers next to polygons are position mean values for these 4 candidate buildings. Background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

4 MODEL TRAINING

4.1 Ground-truth data

In order to have a model generalizable to all US states, we selected three representative types of regions to create ground-truth data from. The three types are medium building density (Nashville TN), high building density (Chicago IL), and mixed building density (Fort Myers FL) regions. We utilized existing building labels on OSM [2] of the three regions for creating ground-truths. To generate ground truth building and address pairs, we start with finding the nearest building for each address geocode from our package scans data and the Geocode1 file, then only keep those with the same building for both types of geocode and agreeing building labels and address texts.

4.2 Training

We break the ground-truth dataset into 75% train (60000 addresses) and 25% test (20000 addresses), then for each address we paired every candidate building to the correct building of an address to compute feature vectors described above. We randomly placed the correct building as the left or the right building in the pairs to create the binary target, which indicates whether the left building is better than the right building for an address. We then trained and tuned a random forest binary classification model with 5-fold stratified cross-validation and selected the best model based on highest accuracy and ROC AUC score on the test set. Finally, to get the BinoML output for an address, we pick the building which wins (determined by a model threshold) over all other candidates in comparison as the best building for that address. Although there are more than 25 features used in the ranking model, the top 10 important features shown in Figure 12 are as expected include some of the major differences among candidate buildings, such as the building's distance to an address's delivery point, the building's name (if available) difference with an address text, the proportion of an address's package scans inside a building, etc.

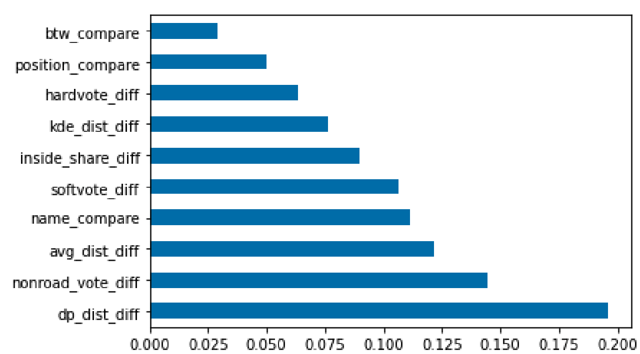

Figure 12: Top 10 features of the trained model.

5 EXPERIMENTAL EVALUATION

Auditor evaluated precision: Using the trained model, we generated BinoML outputs for remaining addresses (not in the ground-truth data) of the 18 months package scans data for each of the three regions. We collaborated with auditors for precision evaluation of these outputs. For each region, auditors randomly selected about 1000 sample outputs and classify each building and address match as true positive (correct match) or false positive (incorrect match) as shown in Table 1. For automatically adding building numbers to building outlines, the precision value needs to be ≥ 99%. Therefore, we decided to use a model threshold of 0.8 for US regions based on the results of Table 1.

Region	Threshold	TPs	FPs	Precision
Nashville TN	0.6	996	4	99.6
Fort Myers FL	0.7	971	7	99.28
Chicago IL	0.7	355	6	98.3
Chicago IL	0.8	1094	6	99.45

Table 1: BinoML sample outputs evaluated by auditors. TPs are true positives and FPs are false positives.

Region	heuristic	BinoML	Multiple
Nashville TN	16878	35128	2.08
Chicago IL	72	343	4.76
Fort Myers FL	4323	15891	3.67

Table 2: Number of unnamed buildings in selected regions of OSM [2] that can be labeled by the heuristic or BinoML with a 0.8 threshold.

Building number coverage: We also computed how many unlabeled buildings in OSM [2] BinoML can add building numbers to. As shown in Table 2, BinoML can label 2 to 5 times more unnamed buildings than the heuristic. In addition, we calculated building number coverage estimate improvement of BinoML with a 0.8 threshold for each region. Since building outlines from OSM often do not indicate their building types (e.g., garage or main building), we can not directly calculate the percentage of labeled buildings among those that should be labeled (garages most likely do not have labels). So for each delivery region, we calculated a BN coverage estimate based on geocodes of addresses. From our previous analysis, except for missing building outlines from the map, most buildings are within 30m of their corresponding addresses' geocodes. Therefore, the BN coverage estimate is the proportion of addresses with any **labeled** building in 30m among addresses with any building in 30m. As shown in Table 3, BinoML can contribute to 12% BN coverage increase even for a region already with > 70% coverage.

False positive anecdotes: Auditors investigated the false positive cases and found the majority of the cases are addresses matched to non-residential buildings next to home buildings as shown in Figure 13. The building highlighted in orange are assigned building numbers from matched addresses. Our auditors think this kind of buildings are not residential buildings since they are in the backside of the plot, so addresses should most likely not matched to them. In terms of areas, this kind of buildings can have similar area to that of a small single family house, so the model may generate false matches when such non-residential buildings are very close to family houses.

Region	OSM	BinoML +OSM	Percentage Increase
Nashville TN	0.7378	0.8271	12.10
Chicago IL	0.9803	0.9814	0.1092
Fort Myers FL	0.8492	0.8668	2.071

Table 3: Building number coverage estimates.

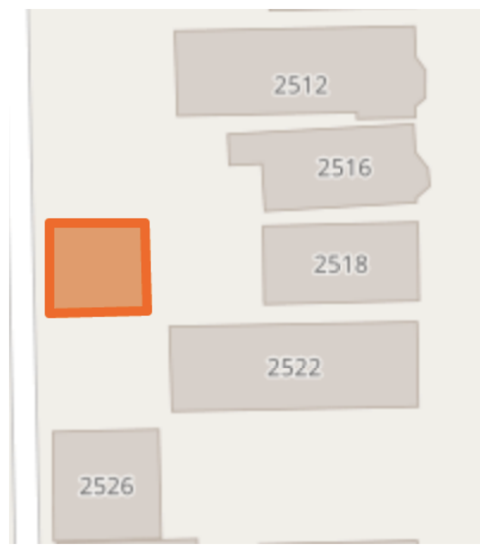

Figure 13: Auditor-provided sample false positive (highlighted) from an urban region. Background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

True positive anecdotes: Sample auditor-evaluated BinoML true positive examples are shown in Figure 14. The available addresses are matched to the correct outlines highlighted in blue and building numbers in address texts are transferred to their matched buildings to label these previously unnamed outlines. In regions with a large area of unlabeled buildings, these new building numbers on a map can serve as helpful visual cues for delivery drivers to know their directions. In addition, we observed many cases where BinoML recovers building labels removed due to low confidence in the heuristic. Figure 15 shows 2 sample correct address and building matches recovered by BinoML.

Figure 14: Sample true positive matches between addresses and buildings (in blue) that result in correct building labels (black numbers). Background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

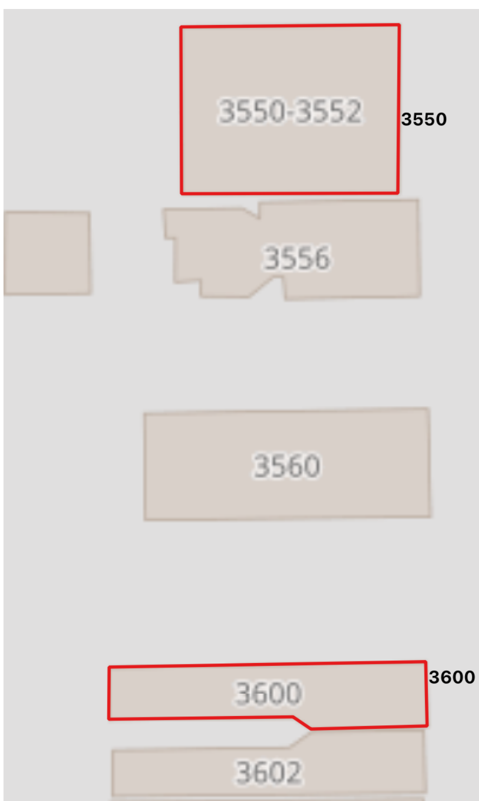

Figure 15: Sample true positive matches that were removed in the heuristic due to low confidence. Bold numbers represent building numbers from address texts. Background: ©OpenStreetMap contributors, CC BY-SA 2.0, link at [2].

6 CONCLUSION

In this paper, we have demonstrated BinoML, a ML ranking model with novelly designed features can automatically add building numbers with high precision to unlabeled buildings by matching customer addresses to correct building outlines. Getting building labels using past delivery scans can save the cost of purchasing such information from third party vendors, which can amount to significant savings once the model expands to more countries. Moreover, with increasing building number coverage from the model's outputs, there is less chance a delivered but not received (DNR) event may happen and more likely improved delivery efficiency due to more labeled buildings as visual cues for delivery drivers. Offline results of the model evaluated using OSM data have shown that coverage improvement in matured geographies (BN coverage > 71%) such as the US can be significant (2%-12%). In currently low BN coverage (< 40%) geographies, the gain is expected to be even greater after fine-tuning the model for a production grade precision in those countries.

DISCLAIMER

Figures with map background have been altered for anonymization.

REFERENCES

[1] Maximum flow problem. https://en.wikipedia.org/wiki/Maximum_flow_problem.
[2] Openstreetmap. https://www.openstreetmap.org/copyright.
[3] F. Chen, N. Chen, H. Mao, and H. Hu. The application of bipartite matching in assignment problem. *CoRR*, abs/1902.00256, 2019.
[4] G. Forman. Getting your package to the right place: Supervised machine learning for geolocation. In *European Conference on Machine Learning (ECML)*, 2021.