

Shortest Path Calculation and Edge-Weight Modeling for Gridded Supply Chain Networks

Jasper Dong¹, Ya Ju Fan², Jean-Paul Watson²

¹University of Washington ²Lawrence Livermore National Laboratory

In supply chain network expansion scenarios - where new nodes are added onto a pre-existing network with weighted edges - one must also determine the edge weights associated with each new candidate node. This project will examine and compare various methods for edge-weight modeling, as well as demonstrate the use of graph algorithms for traversal of supply chain networks.

Introduction

- Supply chain networks are often represented through graph structures, which enables the use of standard graph algorithms for traversal and shortest path calculations
- In network expansion scenarios, new candidate nodes are added onto pre-existing networks, for which associated edge-weights must be determined

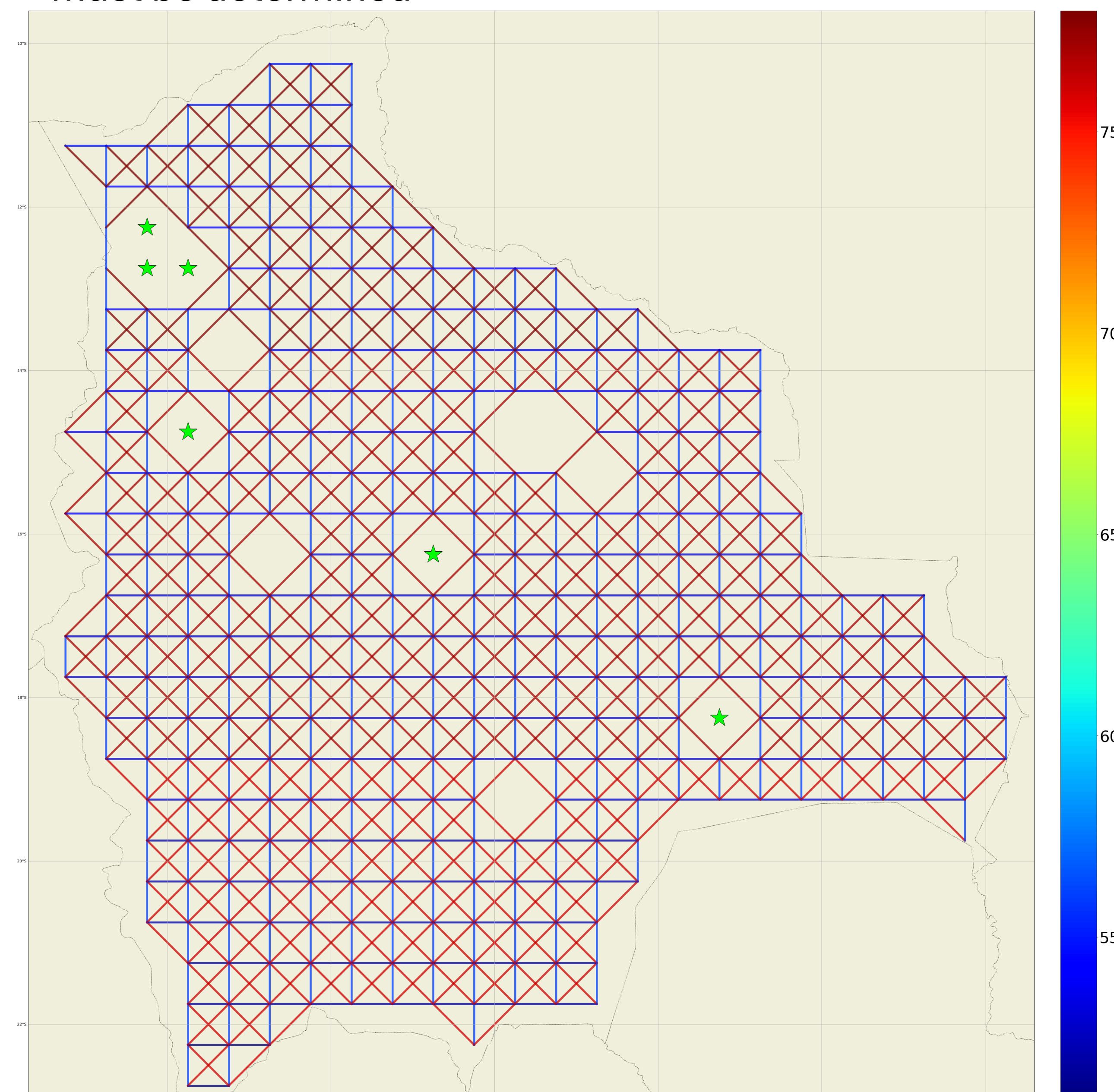


Figure 1. Supply chain network for Bolivia

Scope

In this project, we will compare methods and algorithms across two broad scopes:

Shortest Path Calculation: Comparison of Networkx's **binary heap** structure vs. Scipy's **sparse adjacency matrix** for implementation of Dijkstra's algorithm.

Edge-weight Modeling: Comparison of **deterministic distance calculations** vs. standard machine learning (ML) **prediction models** combined with **nearest neighbor algorithms**.

Shortest Path Calculation

Dijkstra's is used to construct look-up dictionaries of pairwise shortest paths for querying.

Supply node: 262-243
Demand node: 272-250
Weight: 954.0
Path: 262-243 → 271-250 → 272-249 → 272-250

Figure 2. Example shortest path query between two nodes

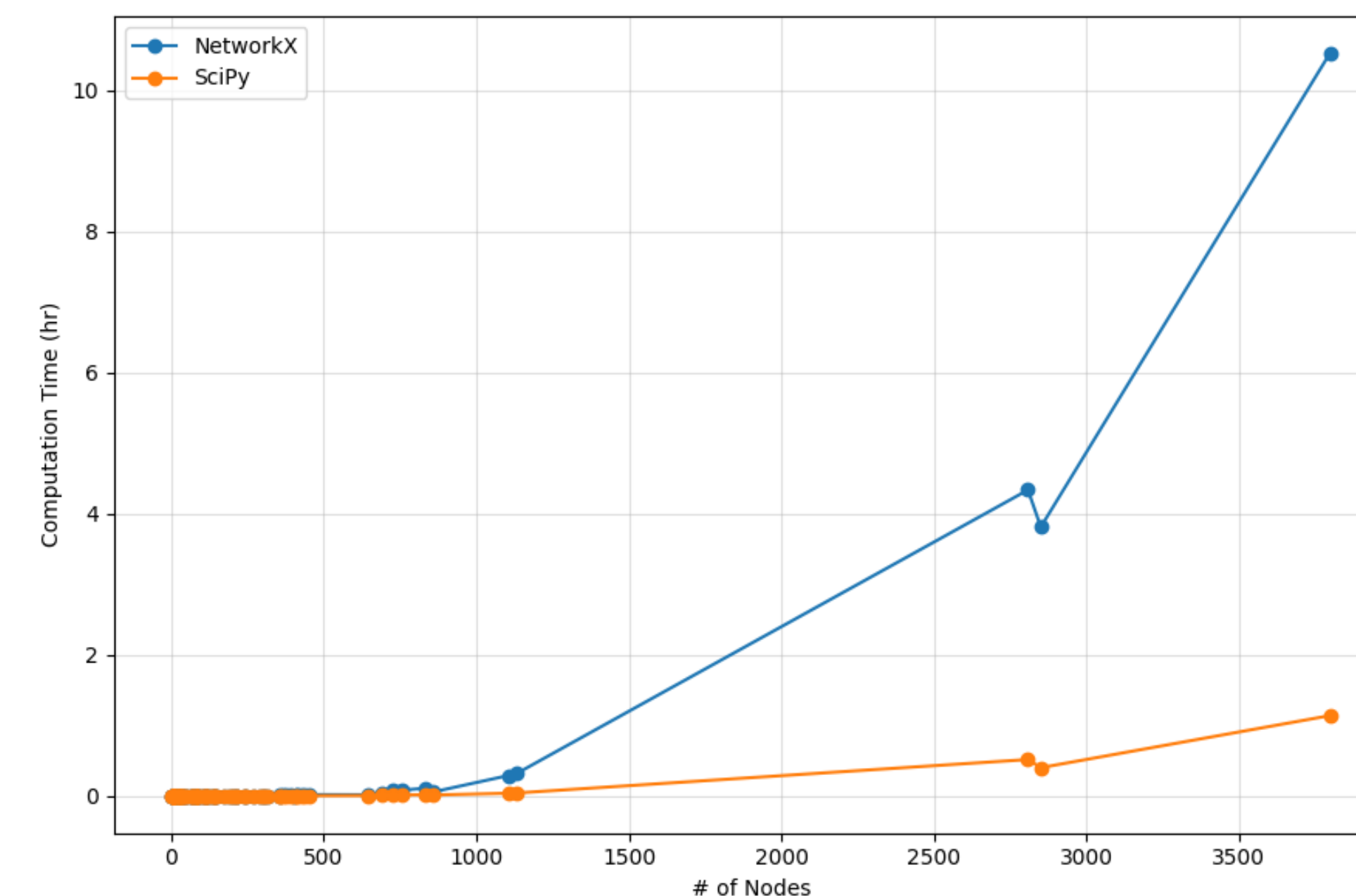


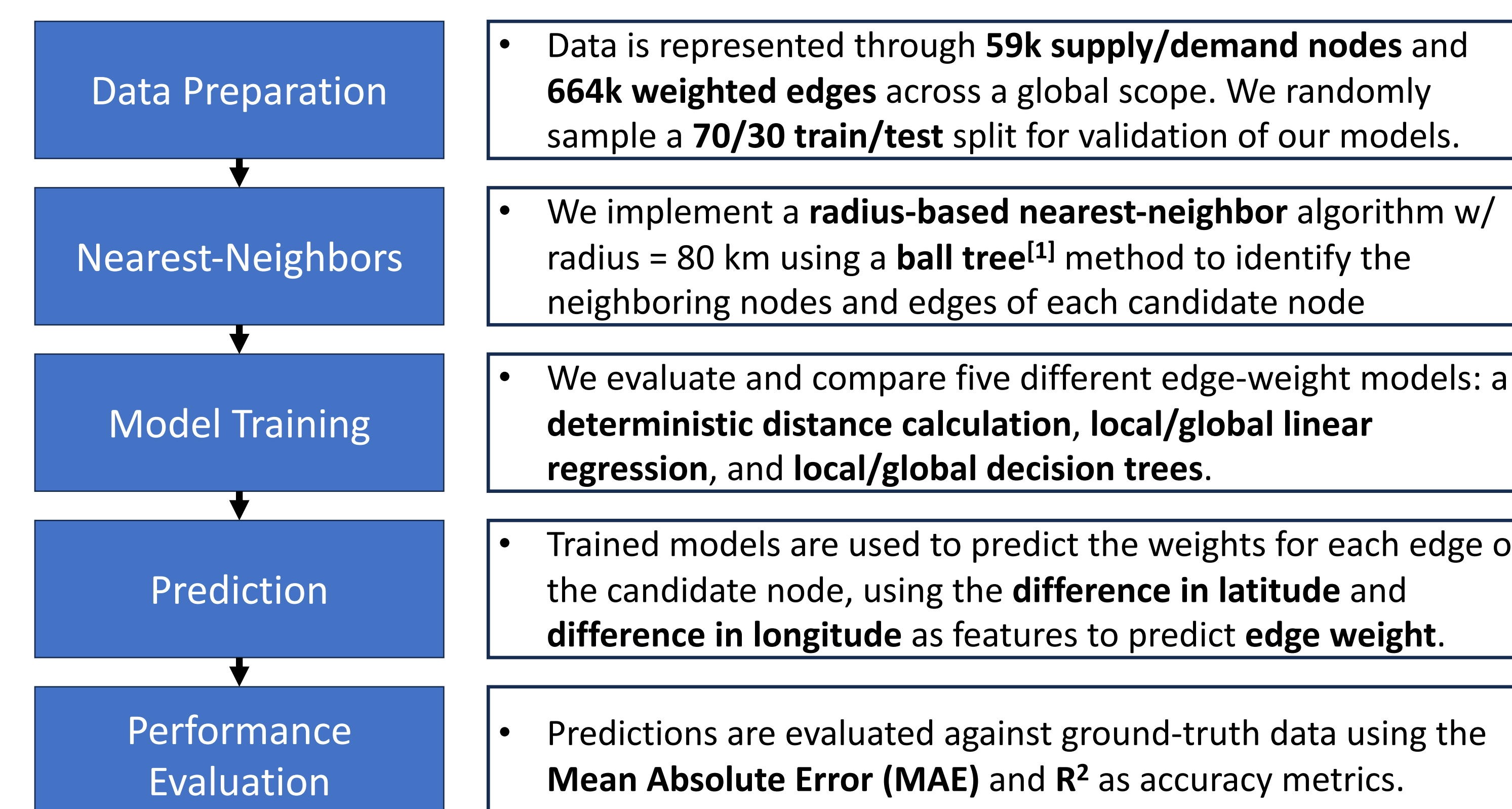
Figure 3. Plot of runtime (hr) vs. # of nodes

Library	Runtime (hr)
Networkx (binary heap)	4.75
Scipy (sparse adj. matrix)	0.38

Table 1. Runtime comparison for building shortest path look-up dictionaries for the USA

Edge-Weight Modeling Experimental Procedure

To compare our edge-weight models, we define the following workflow for assessing a model's performance:



Results

Method	MAE	R2
Haversine Distance	0.0000	1.0000
Local Linear Regression	0.0491	0.9992
Global Linear Regression	4.6328	0.7380
Local Decision Tree	0.0000	1.0000
Global Decision Tree	4.6328	0.7380

Table 2. Accuracy comparison of edge-weight models.

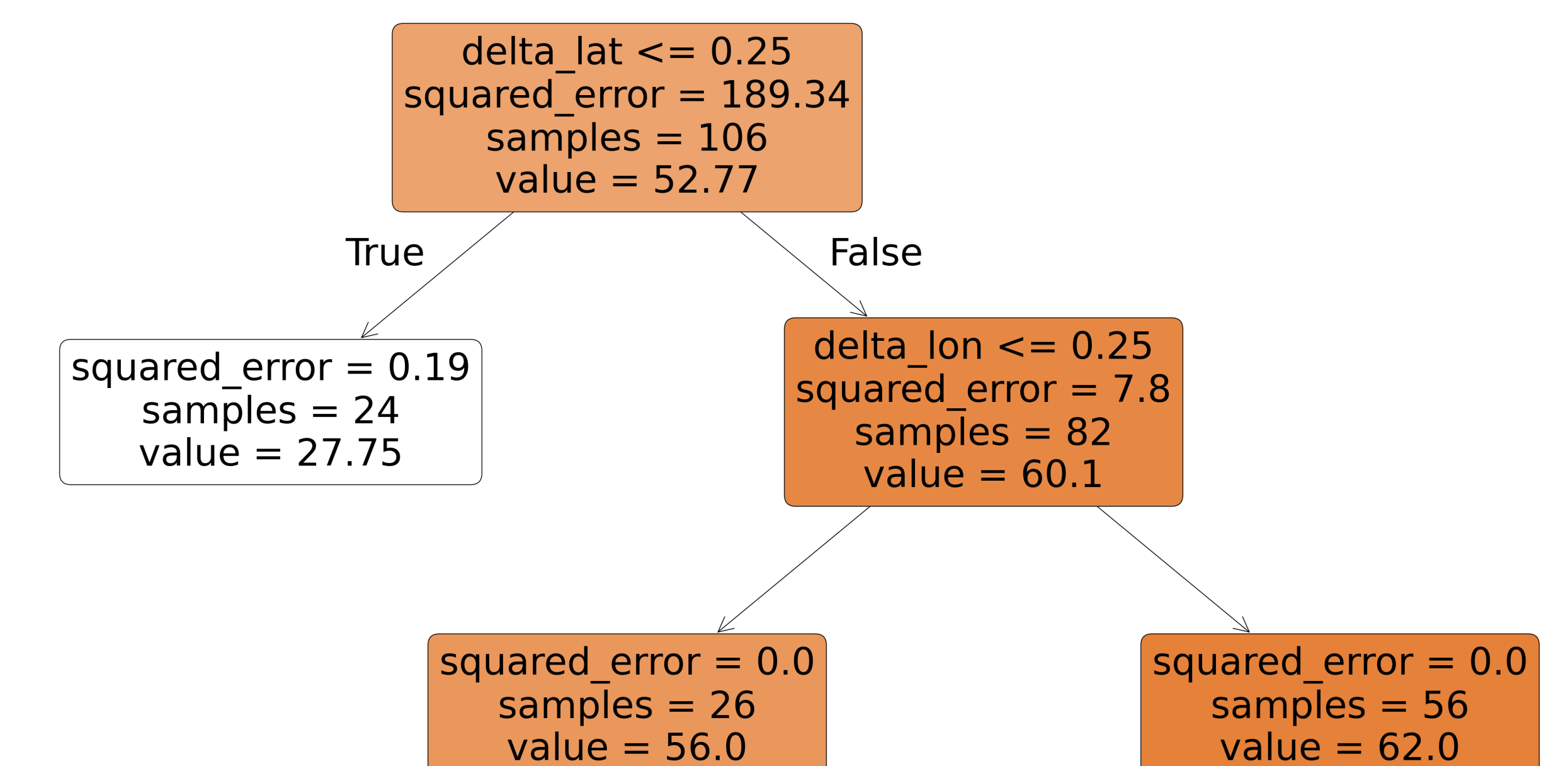


Figure 4. Local decision tree prediction results for example candidate node

Conclusions

- The use of **sparse adjacency matrices** can speed up shortest path calculations compared to traditional data structures.
- Prediction models that are **localized** to their target candidate nodes offer **higher accuracy** compared to global models, due to information for inference being fully contained within a target node's local neighborhood^[2].
- Deterministic formulas for edge-weight modeling may be more efficient at iteratively predicting edge-weights than machine learning (ML) models, especially if candidate edges are nested.