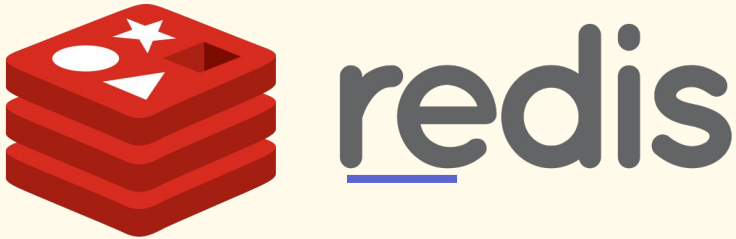
# AGM Data Revolution: Introducing Business Cases for Neo4j, MongoDB, and Redis

Nora, Hannah, and James

# Business Case

Real-Time Location and Charge Tracking for Deliveries




**What is Redis?**

Redis is an open-source, in memory data structure store that can be used as a database, cache and message broker.

**Key Features of Redis**

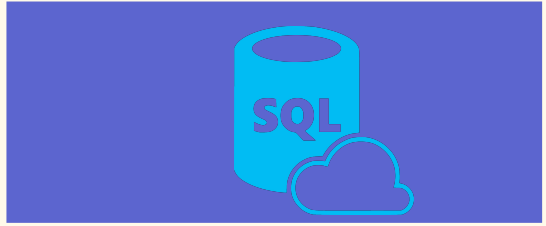
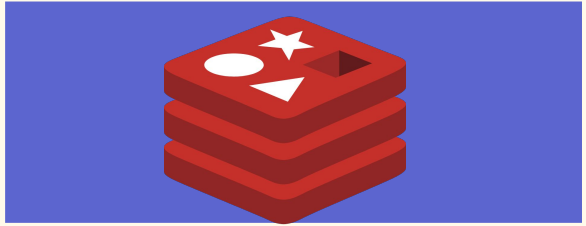
**Speed:** Exceptionally fast read/write operations ideal for real-time applications

**Scalability:** Easily accommodate growing data needs for the mass amount of data needed to operate drones robots and deliveries

**Flexibility**: Suitable for wide range of use cases like caching, session management as well as the ability to support a variety of data types

**Reduced Latency:** Fast application response which improves user experience

# Redis Vs. Traditional Relational Databases

| |  |  |
|---|---|---|
| Real Time Data Handling | Struggles due to disk-based storage | Excels in real-time due to in-memory storage |
| Scalability | Challenging due to complex database management | Highly scalable due to in-memory nature even with large datasets |
| Geospatial Data | Cumbersome and inefficient due SQL requiring more complex processing | Efficiently manages geospatial data and optimized for location tracking |
| Cost -Effectiveness | More Costly due in terms of hardware requirements and scaling complexities. | Lower hardware demands and efficient resource utilization |
| Data Handling for Simple Queries | Requires more structured and detailed query syntax | Simplifies process due to its key-value store structure |

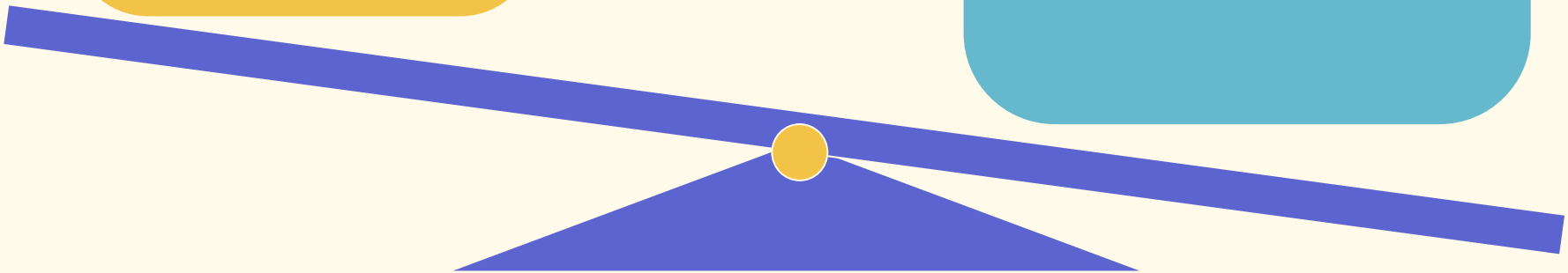
# Costs VS. Benefits of Redis

## Costs

- Initial Setup and Maintenance
- Training for Specialized Skills

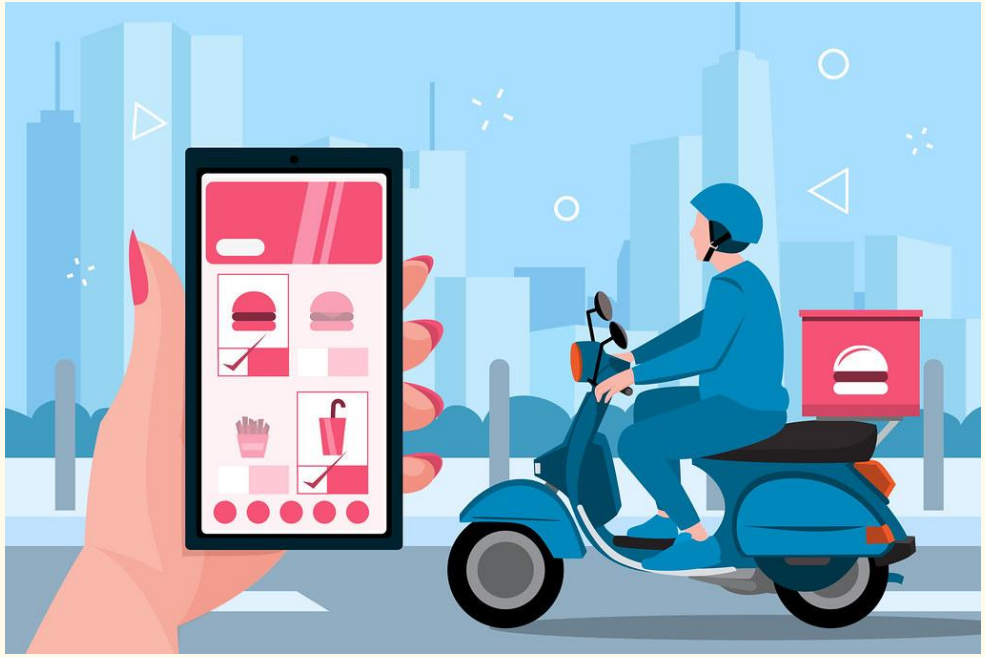
## Benefits

- Increased Efficiency
- Improved Response Time
- Scalability & Flexibility
- Cost-Effectiveness In The Long Run



# Business Case

AGM eventually wants to add several pickup locations for customers at BART stations. They want to analyze details of the pickup locations once they are established.



# What is MongoDB?


MongoDB is a NoSQL Document Database.

Tables:

- Stores
- Customers
- Products

# MongoDB

MongoDB contains collections of JSON Objects

### Store POV #1

- Stores
  - Customers
    - Products

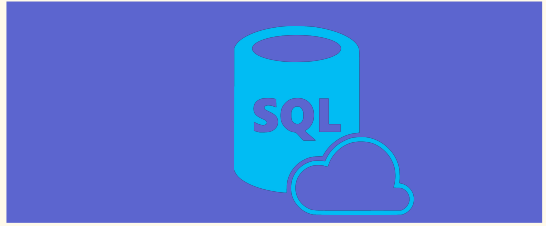
### Customer POV

- Customers
  - Stores
    - Products

### Store POV #2

- Stores
  - Products
    - Customers

# Traditional Relational Databases Vs MongoDB

| |  |  |
|---|---|---|
| Real Time Data Handling | Struggles due to disk-based storage | Has multiple copies of data, needs time to refresh all copies |
| Scalability | Challenging due to complex database management | Highly scalable and can handle large volumes of data |
| Structuring Data | Structure of databases are inflexible | Can create new documents for different structuring of data |
| Cost -Effectiveness | More costly in terms of hardware requirements and scaling complexities. | Cost-efficient for big data and scaling up |
| Querying Data | Get data by querying from several different relations | Can get all data from any created POV because querying pulls documents |

# MongoDB Solution

Pickup Location POV

Pickup Location A
- Store #1
  - Customer 123
  - Customer 456

Pickup Location B
- Store #3
  - Customer 000
  - Customer 346
- Store #10
  - Customer 947
  - Customer 642

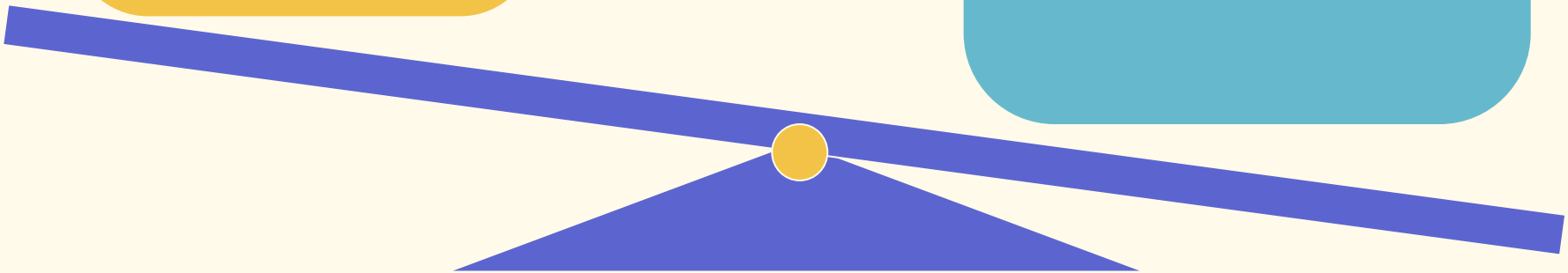
# Costs VS. Benefits of MongoDB

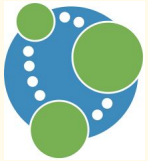
## Costs

- Initial Setup and Maintenance
- Training for Specialized Skills

## Benefits

- Increased Efficiency
- Scalability & Flexibility
- Helpful for analytical purposes





# Neo4j: Business Case

Business case: Choosing a location for our new pickup station

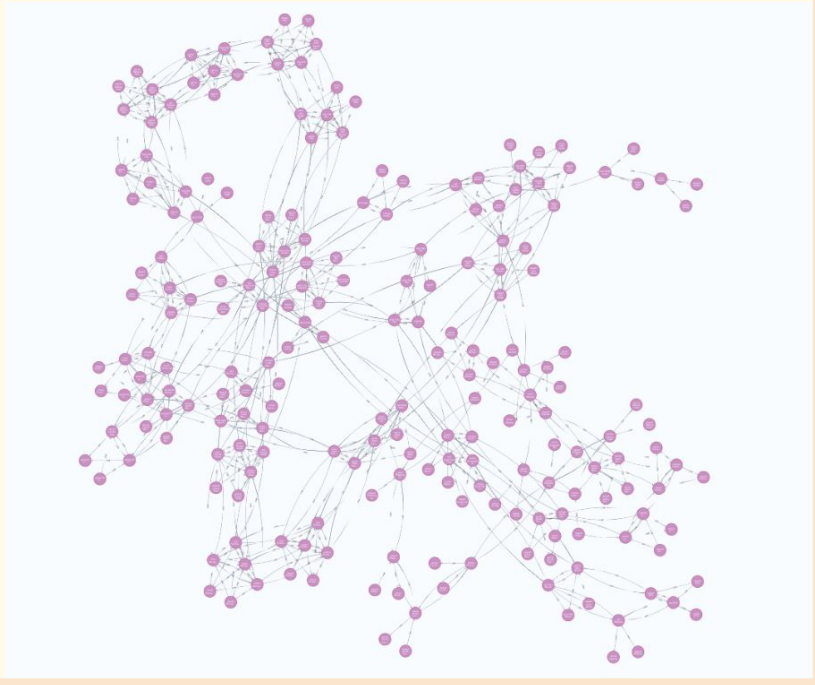
Why choose NoSQL?

- NoSQL databases that support graph structures are better for problems like route optimization than relational database tables. Nodes and paths allow us to run graph algorithms on the data more directly and efficiently.

Benefits of Neo4j:

- Efficiently scale large datasets with billions of connections
- Supports real-time updates to graphs


BART System Stations & Lines



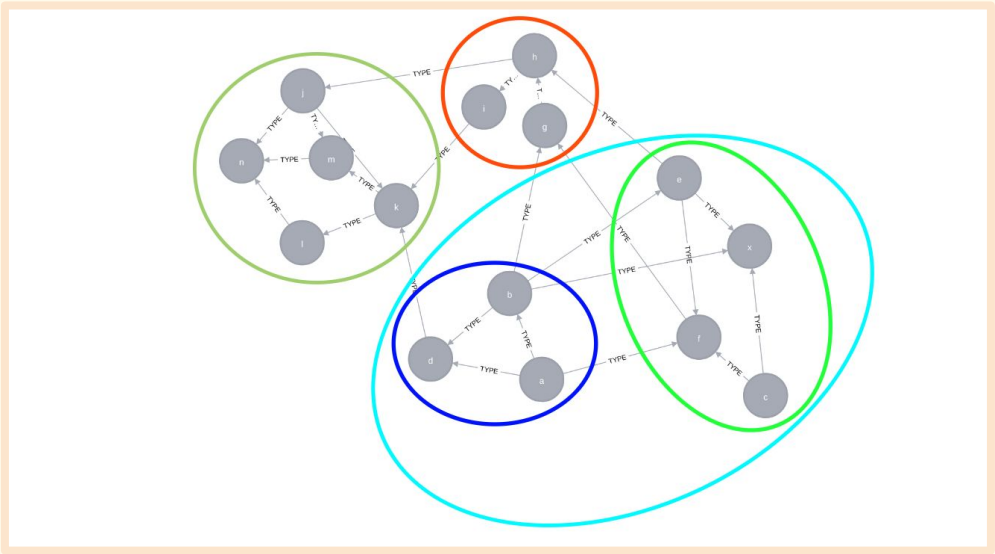
Graphical Representation using Neo4j

# Neo4j Graph Algorithms: Community Detection

## Louvain Modularity

- Definition: A “what if” analysis that tries different groupings of nodes, and tests how well a node fits into a group by comparing relationship weights and densities to an average (or estimate).

- Result: A hierarchy of groups.



# Community Detection Algorithm & BART Daily Exit Data: Choosing New Pickup Locations

| grouped_name | community |
|---|---|
| 16th Street Mission | 118.0 |
| 24th Street Mission | 118.0 |
| Powell Street | 132.0 |
| Civic Center | 132.0 |
| Glen Park | 140.0 |
| Daly City | 140.0 |
| Balboa Park | 140.0 |
| Downtown Berkeley | 142.0 |
| Richmond | 142.0 |
| El Cerrito del Norte | 142.0 |
| Ashby | 142.0 |
| North Berkeley | 142.0 |
| El Cerrito Plaza | 142.0 |

Most Daily Exits

Most Daily Exits

Most Daily Exits

Most Daily Exits

16th St.Mission *(8917 daily exits)*

Powell Street *(14052 daily exits)*

Glen Park *(5425 daily exits)*

Ashby *(5428 daily exits)*

# Business Case

Enhancing Urban Delivery With Shortest Path Algorithm

# Dijkstra's: Pathfinding for Efficient Deliveries

**What is Dijkstra's Algorithm:**

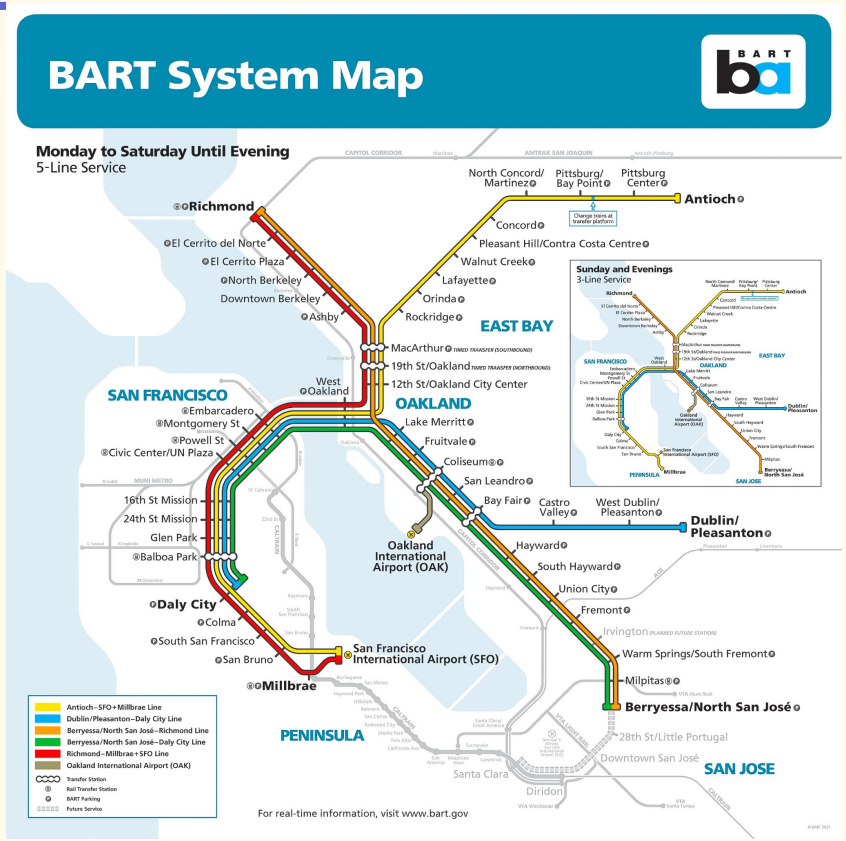
Dijkstra's algorithm is a method for finding the shortest path between nodes in a graph.

**How we will use Dijkstra's Algorithm:**

In the context of our urban delivery system, we want to find the shortest path between these two nodes

1: The customer's closest BART pickup location

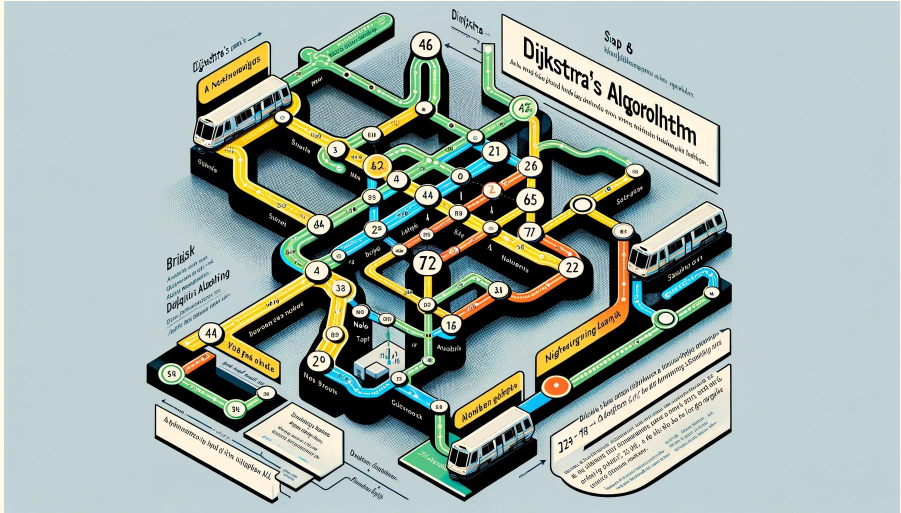
2: The closest BART delivery station to the AGM warehouse/distribution centers



# Implementation

**Input**:

The coordinates of the AGM customer and the Coordinates of the AGM warehouse/distribution center that will deliver the products

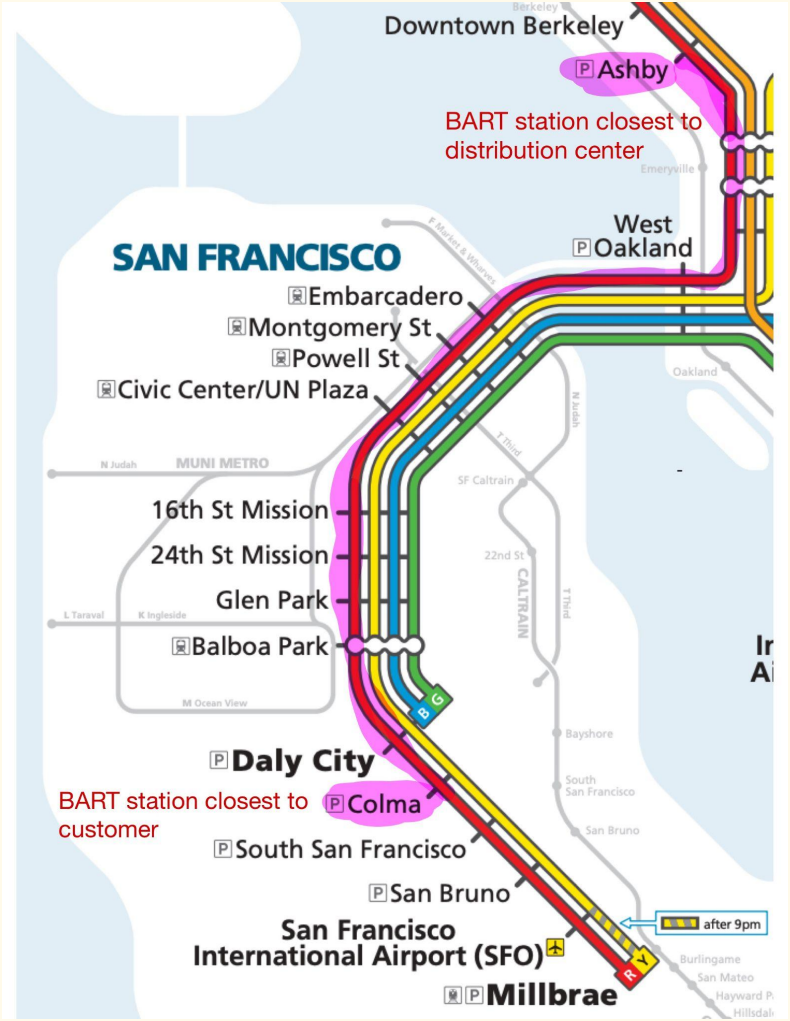



**Output**:

The shortest path via BART between the Customer and the Distribution Center using Dijkstra's which takes into account the travel time between the two nodes

# Example Output

```
[{'Total Cost': 2580,
  'Minutes': 43.0,
  'Path': [('depart Ashby', 0, 0),
   ('red Ashby', 0, 0),
   ('red MacArthur', 240, 240),
   ('red 19th Street', 180, 420),
   ('red 12th Street', 120, 540),
   ('red West Oakland', 300, 840),
   ('red Embarcadero', 420, 1260),
   ('red Montgomery Street', 60, 1320),
   ('red Powell Street', 120, 1440),
   ('red Civic Center', 60, 1500),
   ('red 16th Street Mission', 180, 1680),
   ('red 24th Street Mission', 120, 1800),
   ('red Glen Park', 180, 1980),
   ('red Balboa Park', 120, 2100),
   ('red Daly City', 240, 2340),
   ('red Colma', 240, 2580),
   ('arrive Colma', 0, 2580)]}]
```



# Benefits of Using the Dijkstra's Algorithm

## Enhanced Efficiency

Through shortest and most efficient paths

## Improved Customer Satisfaction

Faster and more reliable deliveries

## Cost Reduction

Optimized routes = lower operational costs

## Real Time Adaptability

Can quickly recalibrate routes based on delays and real time changes

# Closeness Centrality and Pickup Locations

Business Case:

Due to higher crime rates on the east side of the bay, AGM wants to choose a new pickup location on the west side of the BART tracks that can still reach as many bay area customers as possible.

# Closeness Centrality and Pickup Locations

Closeness centrality gives values to each node in the graph based on how close they are to all the other nodes.

Highest closeness: Node with the shortest distance to all other nodes.

# Closeness Centrality to choose Pickup Location

We ran the closeness centrality algorithm to figure out which BART station was the closest to every other BART station so the new pickup location could be easily reached by most of the bay.

Since many BART stations are close together, we grouped them into clusters and averaged their closeness values.

| group | closeness | latitude | longitude |
|---|---|---|---|
| 2 | 0.126380 | 37.790234 | -122.383152 |
| 3 | 0.098247 | 37.742908 | -122.430013 |
| 1 | 0.090022 | 37.893578 | -122.298285 |
| 4 | 0.073094 | 37.661982 | -122.437318 |

# Closeness Centrality to choose Pickup Location

To refine our results, we looked at customer data and calculated how many AGM customers were within 5 miles of groups of BART stations.

```
Customers Reached from stations:
[['Richmond', 'El Cerrito del Norte', 'El Cerrito Plaza', 'North Berkeley', 'Downtown Berkeley', 'Ashby']]
  Customers Reached:  137
```

```
Customers Reached from stations:
[['West Oakland', 'Embarcadero', 'Montgomery Street', 'Powell Street', 'Civic Center']]
  Customers Reached:  149
```

```
Customers Reached from stations:
[['16th Street Mission', '24th Street Mission', 'Glen Park', 'Balboa Park']]
  Customers Reached:  116
```

```
Customers Reached from stations:
[['Daly City', 'Colma', 'South San Francisco', 'San Bruno', 'SFO', 'Milbrae']]
  Customers Reached:  95
```

# Wrap Up

We Covered:

- Neo4j Business Cases
  - Community Detection
  - Dijkstra’s shortest path
  - Closeness Centrality
- Redis Use Cases
- MongoDB Use Cases