# Prometheus

## Device Association Detection for Kiana Systems

John Cutone
Mansi Ganatra
Naga Ritwik Indugu
Shiv Prathik Velagala

# Current Environment

- Kiana Analytics uses intelligent video surveillance to improve security systems and marketing insights for a variety of buildings and events.
- Using triangulation from wifi data allows them to track devices within two meters of accuracy.
- As of now, Kiana’s awareness is limited to the movement of individual devices.

# Idea

- Detect device associations to identify:
  - Devices held by a single person
  - Groups of people who came to the museum together
- Real time and after the fact approach
- Use Cases:
  - Quickly find lost or stolen devices/Persons
  - Create a distinction between group and individual behavior as it applies to security and marketing statistics

# Post data collection plan

- We employ two clustering algorithms, a custom algorithm and SON.
- Our feature vectors will consist of successive longitude, latitude, time points representing an individual device's path.
- We will attempt to cluster/match these paths based on their similarity and use the level of similarity to decide whether a cluster represents a group who entered together or if two devices belong to the same person or not.

# Algorithm 1: Real-time Approach

- Create a grid to hash devices within a certain radius into buckets based on their position.
- Match each pair or group of devices based on their hash and store history of matches.
- Use metrics based on history to declare matched devices.

# Issues with this approach

The data does not come in uniformly, so the first step will require some finesse:

- Create time buffer to try to catch all devices at least once
- Consider all subsets of a hashed grid point
- Use metrics meant to take lack of uniformity into account

# Test Scenario and Acceptance Criteria

- There exists few gaps in the data of various MAC ids that are captured by Kiana analytics.
- Also, the amount of gaps in the data varies.
- This results in missing data and some MAC ids not showing up at a particular timestamp.
- They are randomly caught again by the Kiana systems at a different point of time.
- This creates inconsistencies in the way we can track the location of a device.

# Algorithm 1: Performance Metric

Performance metric: match probability | baseline 98%

Probability (MacId1, MacId2)  =

2 * time matched / (total time(MacId1) + total time(MacId2) )

Weight with Inconsistency constant (MacId1, MacId2):

# of matches / ( total data points (MacId1) + total data points (MacId2) )

Filter by: satisfactory # of data points ~ 1000

# Algorithm 2: End of day Approach

- Implement SON algorithm using Spark
- Using macId’s as key and floor, latitude, longitude, timestamp as values, generate frequent itemsets of lengths > 1
- Experiment with the support threshold

# Algorithm 2: Performance Metric

Visualizations: as no ground truth exists

# End products of the two approaches

- Both plans will output a list, the real-time plan will output a dynamic list and the other a static one.
- This list will include devices matches, whether they're a group or one person, and the strength of that match.
- We will also output statistics like the number of people who entered the museum and group activity.

# Scope

- In-Scope:

1. Clustering devices based on mac addresses, time and location to identify groups of devices together for a threshold period of time.
2. Provide summary statistics based on necessary alterations due to double counted devices.
3. Security assistance for missing devices/persons.

# Scope

- Out-of-Scope:

1. Studying or understanding  group dynamics of the devices which are together- whether the group is family/friends/ colleagues.
2. Technical integration of the solution with existing Kiana Infrastructure.
3. Applying the group insights in areas of marketing and advertising.

# Constraints and Assumptions

Constraints:

1. Dataset has limited features to explore.

Assumptions:

1. Location data shared is triangulated to be accurate up to 2 metres.
2. The group patterns exist.

# Project Timeline

| Key Milestone | Target Date |
|---|---|
| Start Date | 1/17/2019 |
| Define Phase | 1/30/2019 |
| Measure Phase | 2/14/2019 |
| Analyze Phase | 3/31/2019 |
| Improve Phase | 4/15/2019 |
| Control Phase | 4/25/2019 |

# Risks and Dependencies

Risks:

1. Selecting a model that is not accurate
2. False Positives - finding groups that are not actually connected

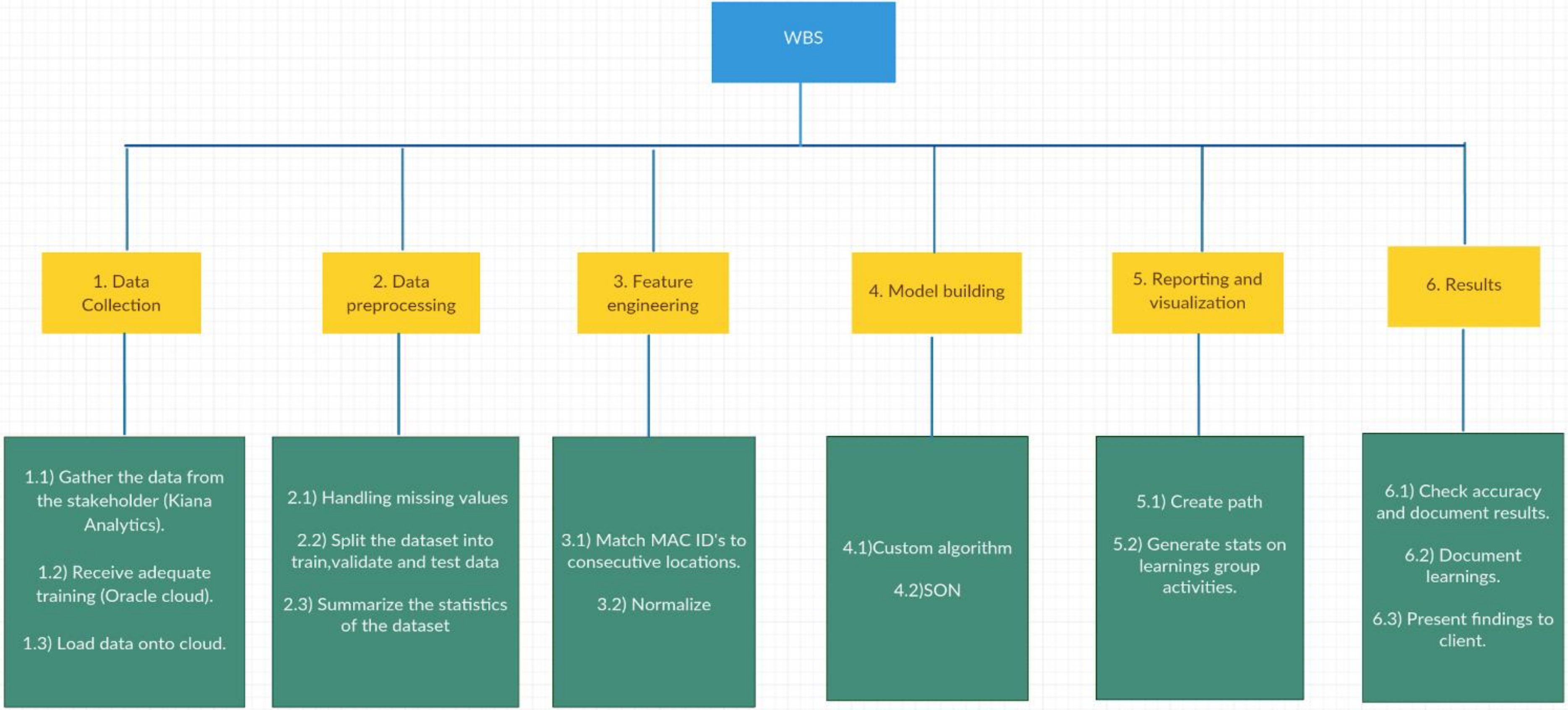
Dependencies:

1. The machines are able to handle the size of the dataset and compute the clusters
2. The data available is enough for training and testing the models

# Risk Management

| Risk Level | Mapped Process | Failure Reasons and Modes | Risk on Failure | Recommendation |
|---|---|---|---|---|
| moderate | Data sorting/cleaning | Imperfect data, gaps in timeline | Unable to implement algorithm that relies on continuous data | Create multiple algorithms, some must be capable of handling imperfect data |
| high | Model Selection | Overfitting, poor choice of accuracy measure | Project deliverable is not helpful to client | Use different accuracy measure and visualizations to check for correctness |
| moderate | Generating device associations | Underestimate device proximity of separate visitors | False positives leading to inaccurate results | Visualize paths of generated matches to check for accuracy |
| low | Performing calculations for device associations | High computation time takes away from real-time aspect of solution | Cannot implement solution in real time | Create separate solution meant for after-the-fact implementation |

# WBS Diagram



WBS

1. Data Collection

1.1) Gather the data from the stakeholder (Kiana Analytics).

1.2) Receive adequate training (Oracle cloud).

1.3) Load data onto cloud.

2. Data preprocessing

2.1) Handling missing values

2.2) Split the dataset into train,validate and test data

2.3) Summarize the statistics of the dataset

3. Feature engineering

3.1) Match MAC ID's to consecutive locations.

3.2) Normalize

4. Model building

4.1)Custom algorithm

4.2)SON

5. Reporting and visualization

5.1) Create path

5.2) Generate stats on learnings group activities.

6. Results

6.1) Check accuracy and document results.

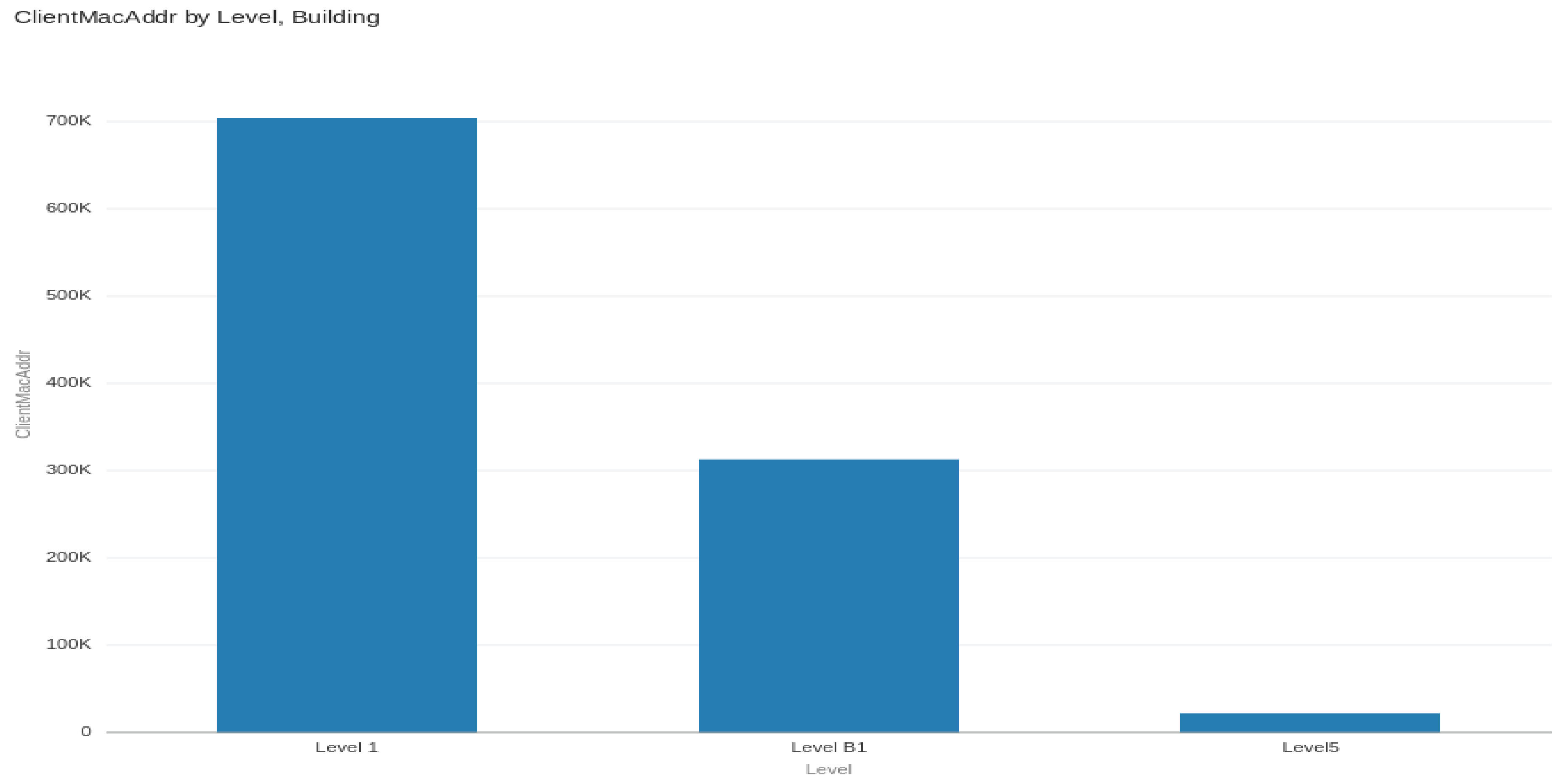
6.2) Document learnings.

6.3) Present findings to client.

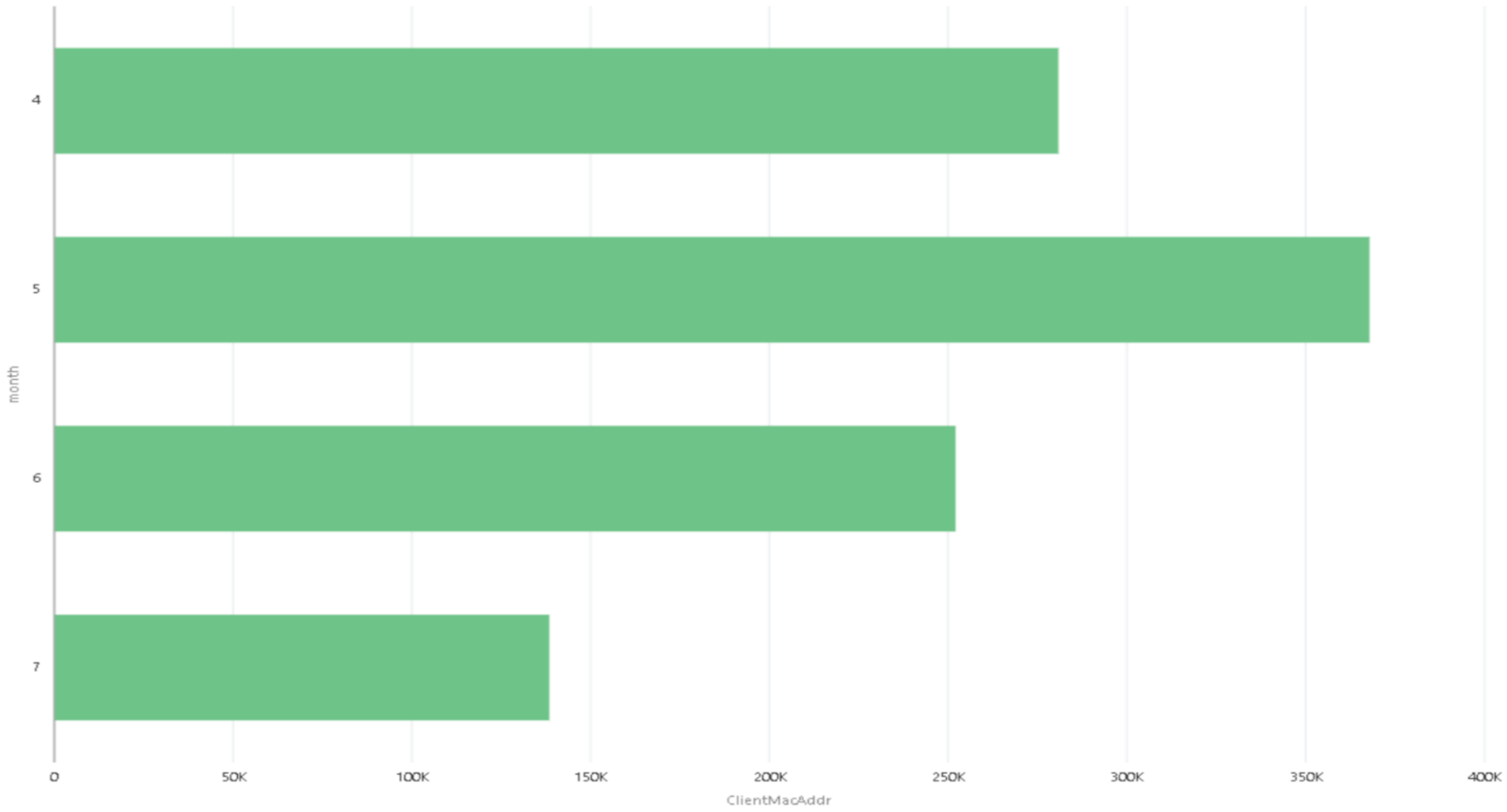
# Sampling of Data

- For EDA purpose we used 1 million records which were randomly sampled from the entire dataset.  Later we understood that the sample generated wasn't useful for our Algorithm as we needed data for single days.
- For the Algorithms we have taken all the data for date 05/12/2018 which has ~ 3,653,932
- Another date we considered was 06/06/20218 which has ~ 5,114,632
- The dates were chosen randomly.

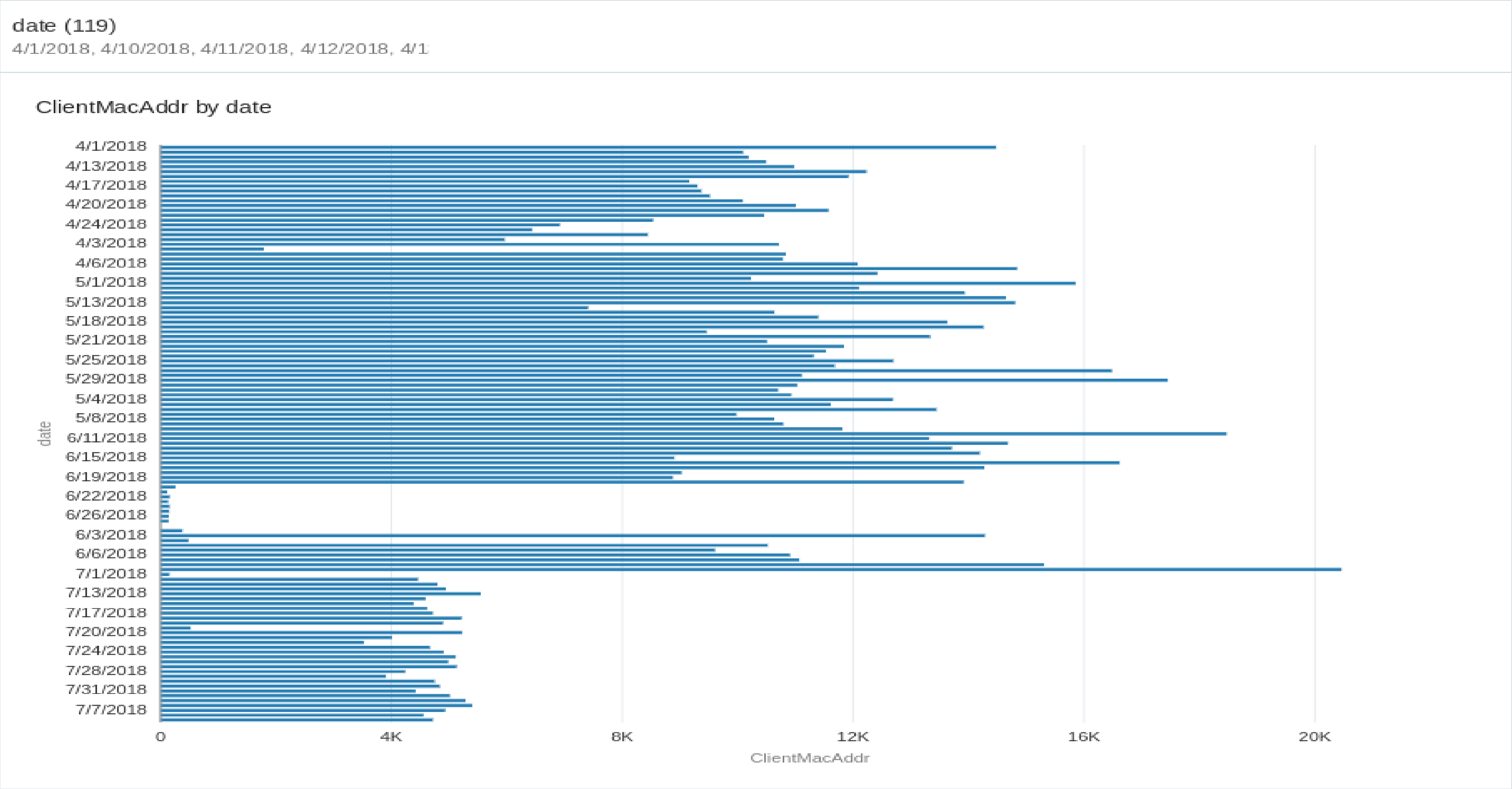
# Exploratory Data Analysis



ClientMacAddr by month



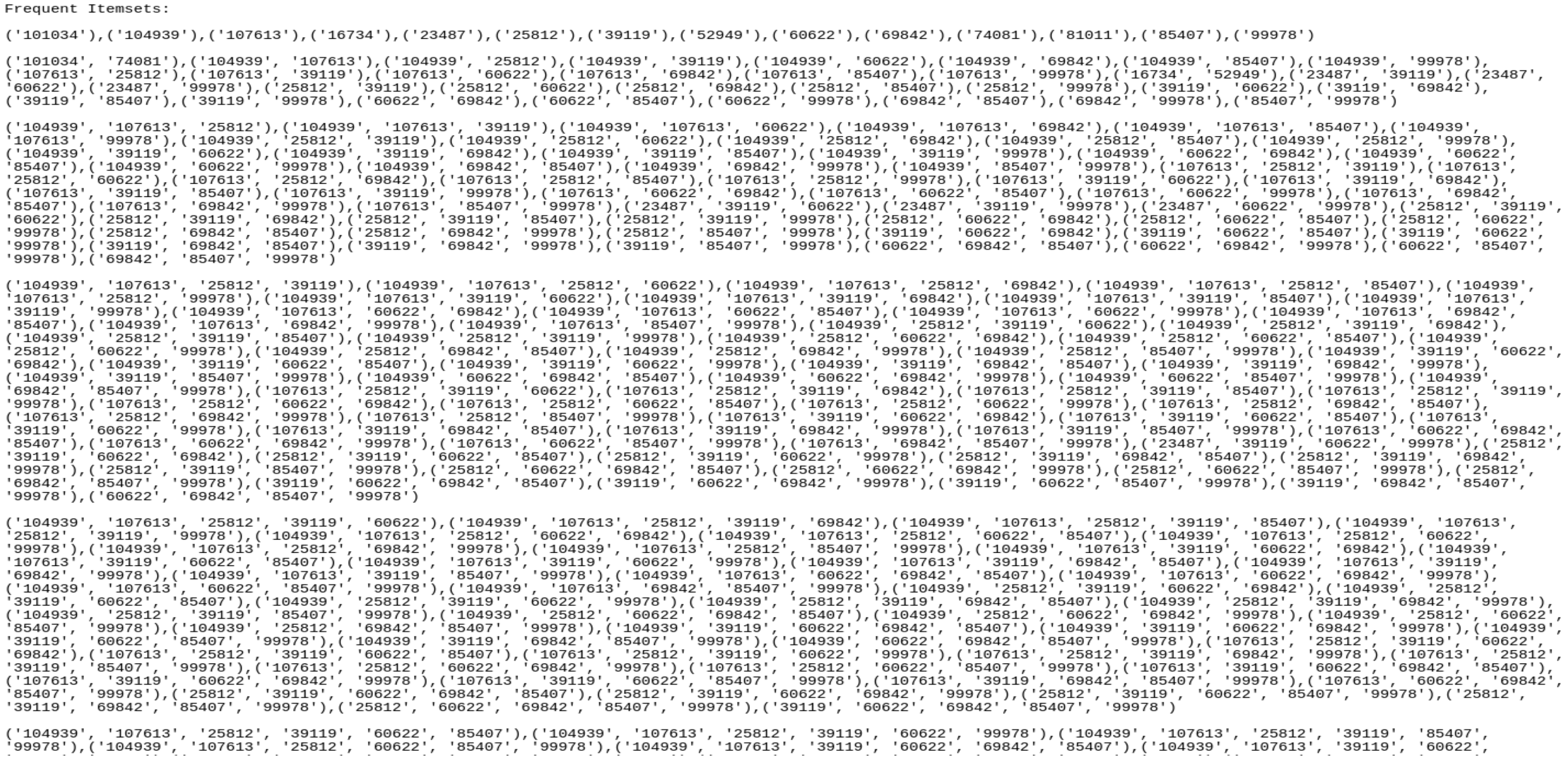
# EDA Cont...



date (119)

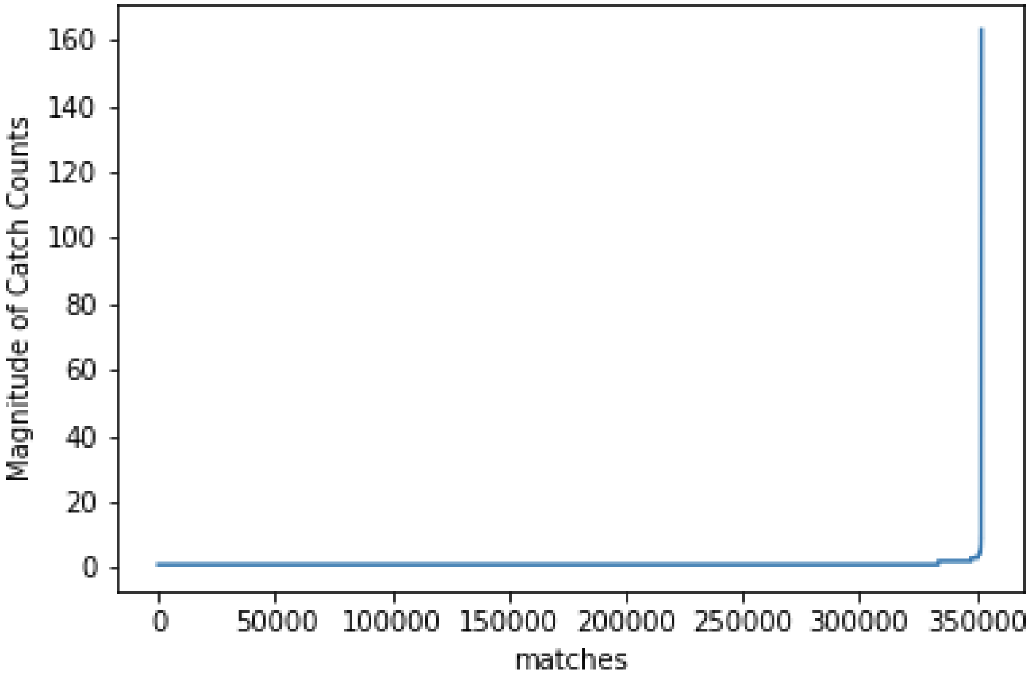
4/1/2018, 4/10/2018, 4/11/2018, 4/12/2018, 4/1

# SON results



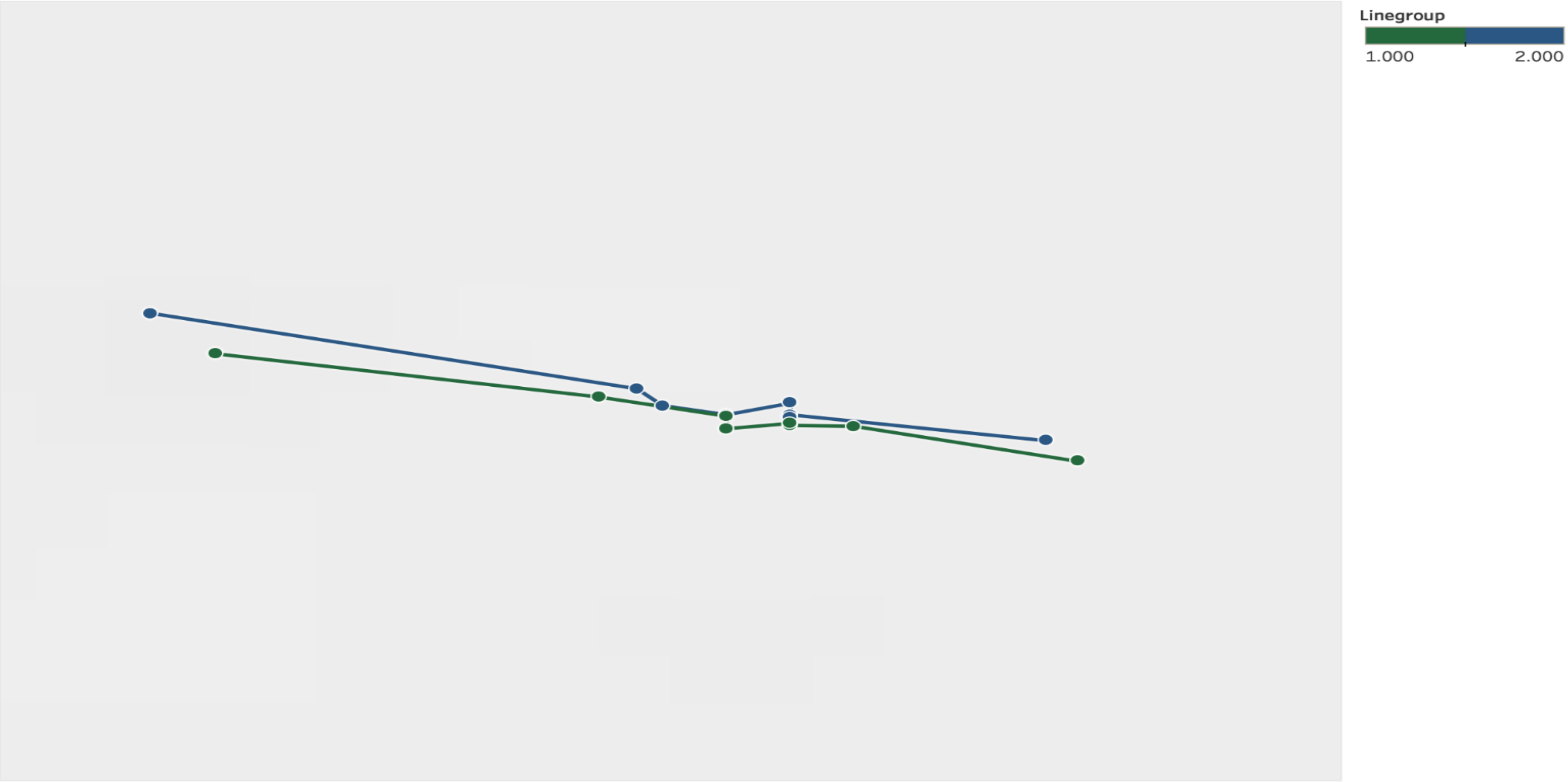
```
Frequent Itemsets:

('101034'),('104939'),('107613'),('16734'),('23487'),('25812'),('39119'),('52949'),('60622'),('69842'),('74081'),('81011'),('85407'),('99978')

('101034', '74081'),('104939', '107613'),('104939', '25812'),('104939', '39119'),('104939', '60622'),('104939', '69842'),('104939', '85407'),('104939', '99978'),
('107613', '25812'),('107613', '39119'),('107613', '60622'),('107613', '69842'),('107613', '85407'),('107613', '99978'),('16734', '52949'),('23487', '39119'),('23487',
'60622'),('23487', '99978'),('25812', '39119'),('25812', '60622'),('25812', '69842'),('25812', '85407'),('25812', '99978'),('39119', '60622'),('39119', '69842'),
('39119', '85407'),('39119', '99978'),('60622', '69842'),('60622', '85407'),('60622', '99978'),('69842', '85407'),('69842', '99978'),('85407', '99978')

('104939', '107613', '25812'),('104939', '107613', '39119'),('104939', '107613', '60622'),('104939', '107613', '69842'),('104939', '107613', '85407'),('104939',
'107613', '99978'),('104939', '25812', '39119'),('104939', '25812', '60622'),('104939', '25812', '69842'),('104939', '25812', '85407'),('104939', '25812', '99978'),
('104939', '39119', '60622'),('104939', '39119', '69842'),('104939', '39119', '85407'),('104939', '39119', '99978'),('104939', '60622', '69842'),('104939', '60622',
'85407'),('104939', '60622', '99978'),('104939', '69842', '85407'),('104939', '69842', '99978'),('104939', '85407', '99978'),('107613', '25812', '39119'),('107613',
'25812', '60622'),('107613', '25812', '69842'),('107613', '25812', '85407'),('107613', '25812', '99978'),('107613', '39119', '60622'),('107613', '39119', '69842'),
('107613', '39119', '85407'),('107613', '39119', '99978'),('107613', '60622', '69842'),('107613', '60622', '85407'),('107613', '60622', '99978'),('107613', '69842',
'85407'),('107613', '69842', '99978'),('107613', '85407', '99978'),('23487', '39119', '60622'),('23487', '39119', '99978'),('23487', '60622', '99978'),('25812', '39119',
'60622'),('25812', '39119', '69842'),('25812', '39119', '85407'),('25812', '39119', '99978'),('25812', '60622', '69842'),('25812', '60622', '85407'),('25812', '60622',
'99978'),('25812', '69842', '85407'),('25812', '69842', '99978'),('25812', '85407', '99978'),('39119', '60622', '69842'),('39119', '60622', '85407'),('39119', '60622',
'99978'),('39119', '69842', '85407'),('39119', '69842', '99978'),('39119', '85407', '99978'),('60622', '69842', '85407'),('60622', '69842', '99978'),('60622', '85407',
'99978'),('69842', '85407', '99978')

('104939', '107613', '25812', '39119'),('104939', '107613', '25812', '60622'),('104939', '107613', '25812', '69842'),('104939', '107613', '25812', '85407'),('104939',
'107613', '25812', '99978'),('104939', '107613', '39119', '60622'),('104939', '107613', '39119', '69842'),('104939', '107613', '39119', '85407'),('104939', '107613',
'39119', '99978'),('104939', '107613', '60622', '69842'),('104939', '107613', '60622', '85407'),('104939', '107613', '60622', '99978'),('104939', '107613', '69842',
'85407'),('104939', '107613', '69842', '99978'),('104939', '107613', '85407', '99978'),('104939', '25812', '39119', '60622'),('104939', '25812', '39119', '69842'),
('104939', '25812', '39119', '85407'),('104939', '25812', '39119', '99978'),('104939', '25812', '60622', '69842'),('104939', '25812', '60622', '85407'),('104939',
'25812', '60622', '99978'),('104939', '25812', '69842', '85407'),('104939', '25812', '69842', '99978'),('104939', '25812', '85407', '99978'),('104939', '39119', '60622',
'69842'),('104939', '39119', '60622', '85407'),('104939', '39119', '60622', '99978'),('104939', '39119', '69842', '85407'),('104939', '39119', '69842', '99978'),
('104939', '39119', '85407', '99978'),('104939', '60622', '69842', '85407'),('104939', '60622', '69842', '99978'),('104939', '60622', '85407', '99978'),('104939',
'69842', '85407', '99978'),('107613', '25812', '39119', '60622'),('107613', '25812', '39119', '69842'),('107613', '25812', '39119', '85407'),('107613', '25812', '39119',
'99978'),('107613', '25812', '60622', '69842'),('107613', '25812', '60622', '85407'),('107613', '25812', '60622', '99978'),('107613', '25812', '69842', '85407'),
('107613', '25812', '69842', '99978'),('107613', '25812', '85407', '99978'),('107613', '39119', '60622', '69842'),('107613', '39119', '60622', '85407'),('107613',
'39119', '60622', '99978'),('107613', '39119', '69842', '85407'),('107613', '39119', '69842', '99978'),('107613', '39119', '85407', '99978'),('107613', '60622', '69842',
'85407'),('107613', '60622', '69842', '99978'),('107613', '60622', '85407', '99978'),('107613', '69842', '85407', '99978'),('23487', '39119', '60622', '99978'),('25812',
'39119', '60622', '69842'),('25812', '39119', '60622', '85407'),('25812', '39119', '60622', '99978'),('25812', '39119', '69842', '85407'),('25812', '39119', '69842',
'99978'),('25812', '39119', '85407', '99978'),('25812', '60622', '69842', '85407'),('25812', '60622', '69842', '99978'),('25812', '60622', '85407', '99978'),('25812',
'69842', '85407', '99978'),('39119', '60622', '69842', '85407'),('39119', '60622', '69842', '99978'),('39119', '60622', '85407', '99978'),('39119', '69842', '85407',
'99978'),('60622', '69842', '85407', '99978')

('104939', '107613', '25812', '39119', '60622'),('104939', '107613', '25812', '39119', '69842'),('104939', '107613', '25812', '39119', '85407'),('104939', '107613',
'25812', '39119', '99978'),('104939', '107613', '25812', '60622', '69842'),('104939', '107613', '25812', '60622', '85407'),('104939', '107613', '25812', '60622',
'99978'),('104939', '107613', '25812', '69842', '99978'),('104939', '107613', '25812', '85407', '99978'),('104939', '107613', '39119', '60622', '69842'),('104939',
'107613', '39119', '60622', '85407'),('104939', '107613', '39119', '60622', '99978'),('104939', '107613', '39119', '69842', '85407'),('104939', '107613', '39119',
'69842', '99978'),('104939', '107613', '39119', '85407', '99978'),('104939', '107613', '60622', '69842', '85407'),('104939', '107613', '60622', '69842', '99978'),
('104939', '107613', '60622', '85407', '99978'),('104939', '107613', '69842', '85407', '99978'),('104939', '25812', '39119', '60622', '69842'),('104939', '25812',
'39119', '60622', '85407'),('104939', '25812', '39119', '60622', '99978'),('104939', '25812', '39119', '69842', '85407'),('104939', '25812', '39119', '69842', '99978'),
('104939', '25812', '39119', '85407', '99978'),('104939', '25812', '60622', '69842', '85407'),('104939', '25812', '60622', '69842', '99978'),('104939', '25812', '60622',
'85407', '99978'),('104939', '25812', '69842', '85407', '99978'),('104939', '39119', '60622', '69842', '85407'),('104939', '39119', '60622', '69842', '99978'),('104939',
'39119', '60622', '85407', '99978'),('104939', '39119', '69842', '85407', '99978'),('104939', '60622', '69842', '85407', '99978'),('107613', '25812', '39119', '60622',
'69842'),('107613', '25812', '39119', '60622', '85407'),('107613', '25812', '39119', '60622', '99978'),('107613', '25812', '39119', '69842', '99978'),('107613', '25812',
'39119', '85407', '99978'),('107613', '25812', '60622', '69842', '99978'),('107613', '25812', '60622', '85407', '99978'),('107613', '39119', '60622', '69842', '85407'),
('107613', '39119', '60622', '69842', '99978'),('107613', '39119', '60622', '85407', '99978'),('107613', '39119', '69842', '85407', '99978'),('107613', '60622', '69842',
'85407', '99978'),('25812', '39119', '60622', '69842', '85407'),('25812', '39119', '60622', '69842', '99978'),('25812', '39119', '60622', '85407', '99978'),('25812',
'39119', '69842', '85407', '99978'),('25812', '60622', '69842', '85407', '99978'),('39119', '60622', '69842', '85407', '99978')

('104939', '107613', '25812', '39119', '60622', '85407'),('104939', '107613', '25812', '39119', '60622', '99978'),('104939', '107613', '25812', '39119', '85407',
'99978'),('104939', '107613', '25812', '60622', '85407', '99978'),('104939', '107613', '39119', '60622', '69842', '85407'),('104939', '107613', '39119', '60622',
```

# Match count distribution



# Visualizations

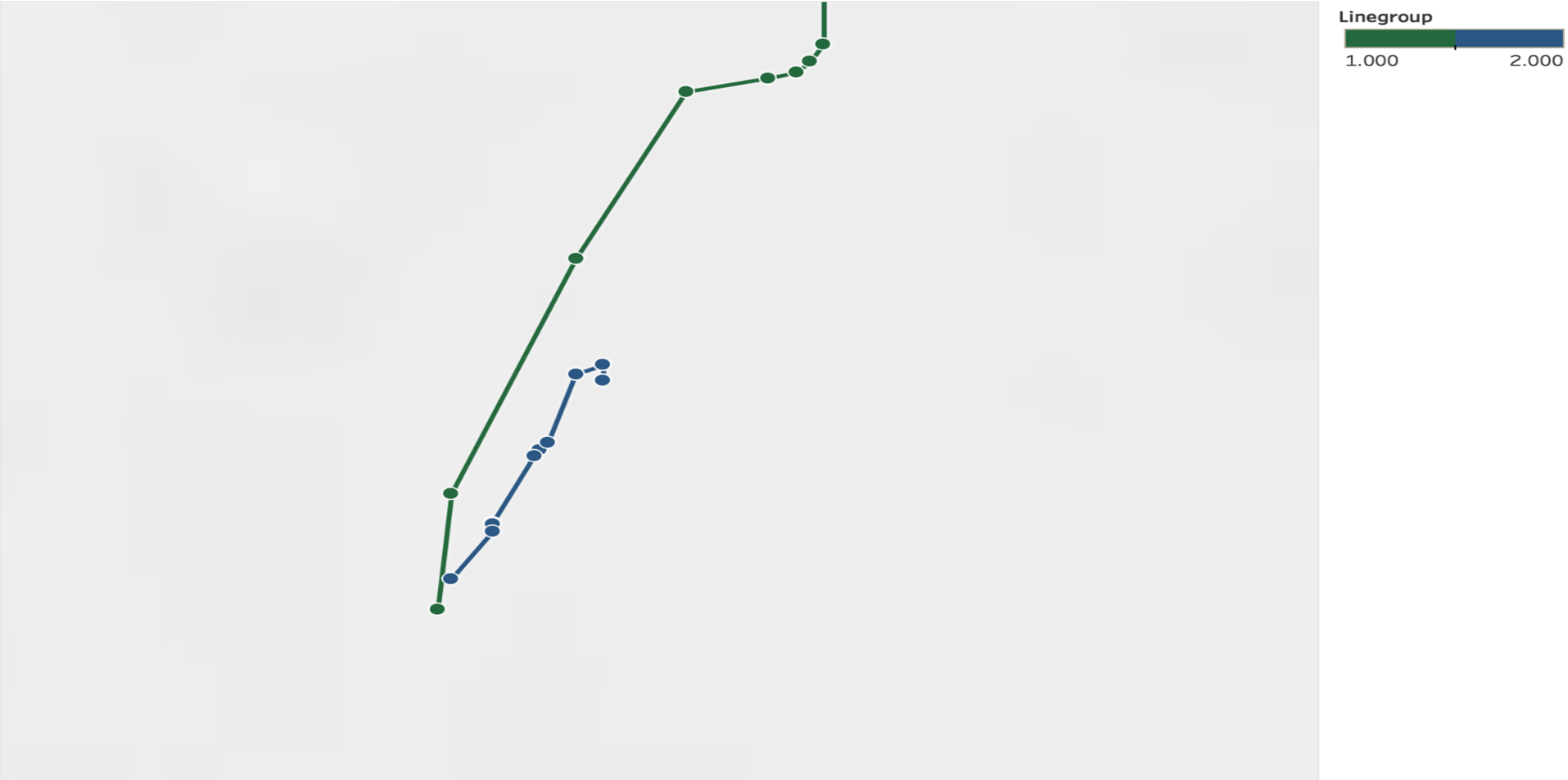
<Two minute period, high match>



Map based on average of Long and average of Long and average of Lat. Color shows details about Linegroup. Details are shown for Floor and Time. The view is filtered on Time, which ranges from 5/12/2018 11:15:46 AM to 5/12/2018 11:17:00 AM.

# Visualizations

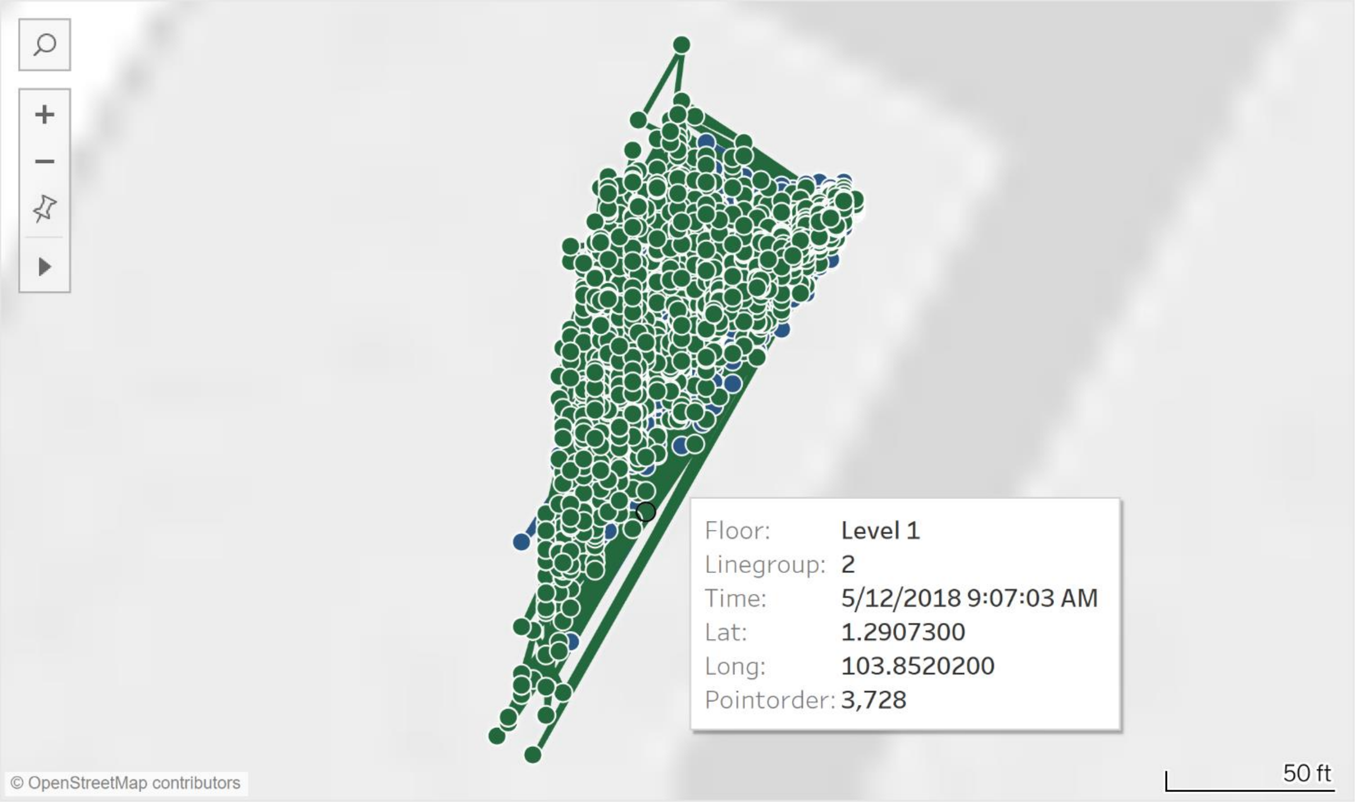
<Two minute period, only 93% match>



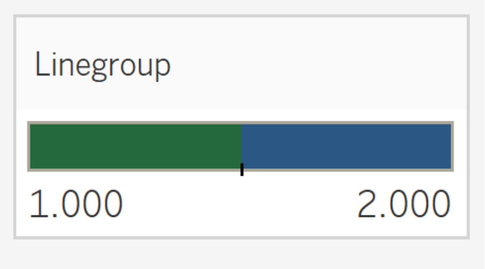
Map based on average of Long and average of Long and average of Lat. Color shows details about Linegroup. Details are shown for Floor and Time. The view is filtered on Time, which ranges from 5/12/2018 7:17:31 PM to 5/12/2018 7:20:08 PM.

# Visualizations

Path tracking of 2 devices example



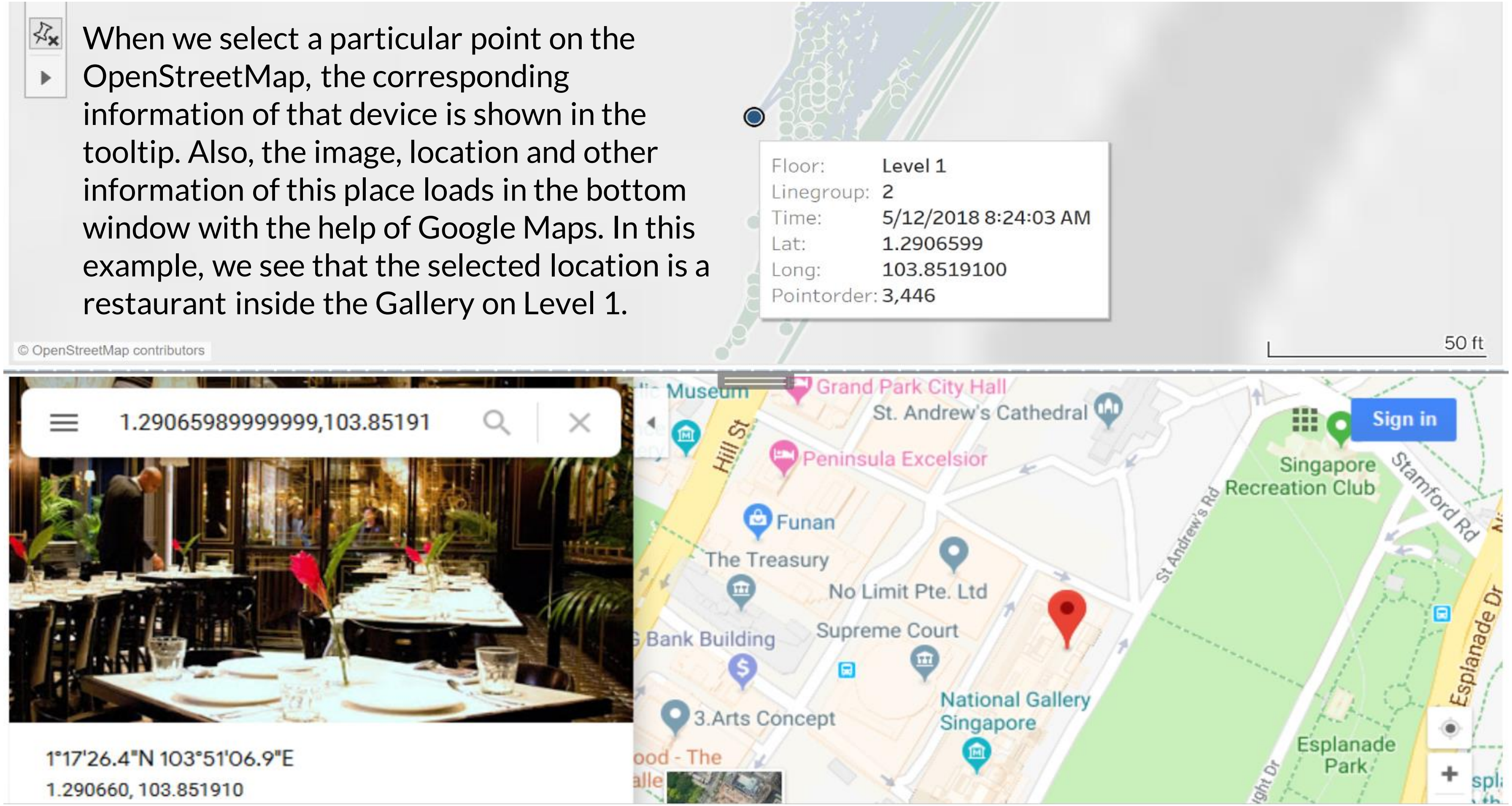
Floor: Level 1
Linegroup: 2
Time: 5/12/2018 9:07:03 AM
Lat: 1.2907300
Long: 103.8520200
Pointorder: 3,728



This is an example showing the locations of 2 devices at various timestamps. We can observe that the path followed by them is almost identical to each other and these two devices can be clustered together as belonging to one group.

# Visualization example using OpenStreetMap & Google Maps

When we select a particular point on the OpenStreetMap, the corresponding information of that device is shown in the tooltip. Also, the image, location and other information of this place loads in the bottom window with the help of Google Maps. In this example, we see that the selected location is a restaurant inside the Gallery on Level 1.

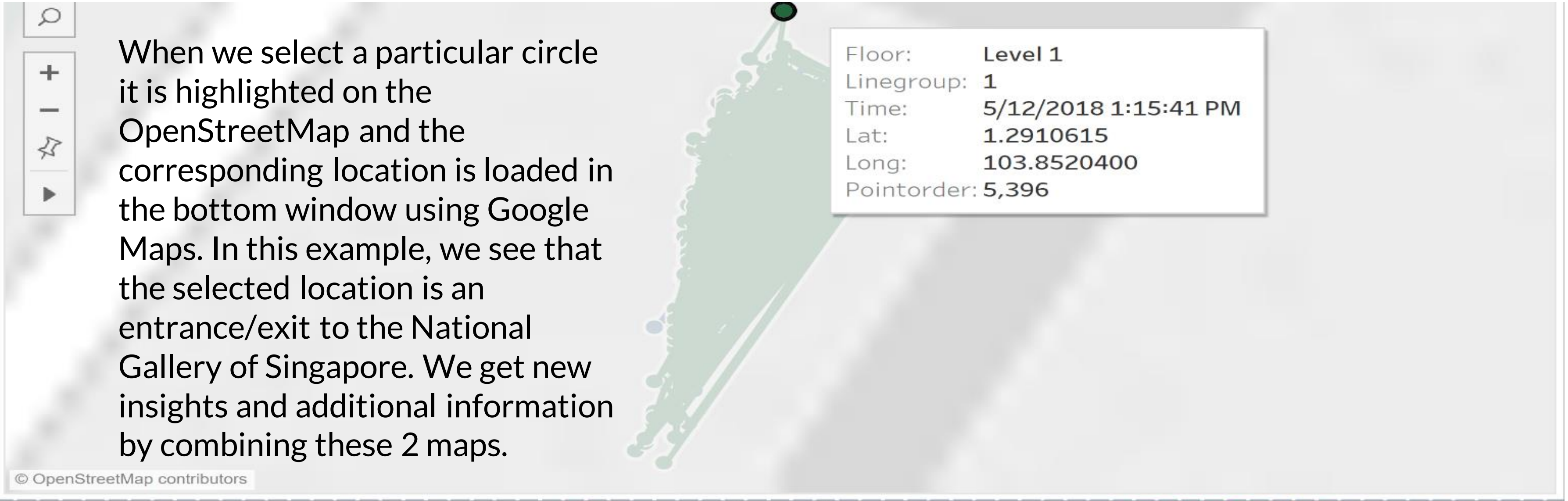


| | |
|---|---|
| Floor: | Level 1 |
| Linegroup: | 2 |
| Time: | 5/12/2018 8:24:03 AM |
| Lat: | 1.2906599 |
| Long: | 103.8519100 |
| Pointorder: | 3,446 |

1.29065989999999,103.85191

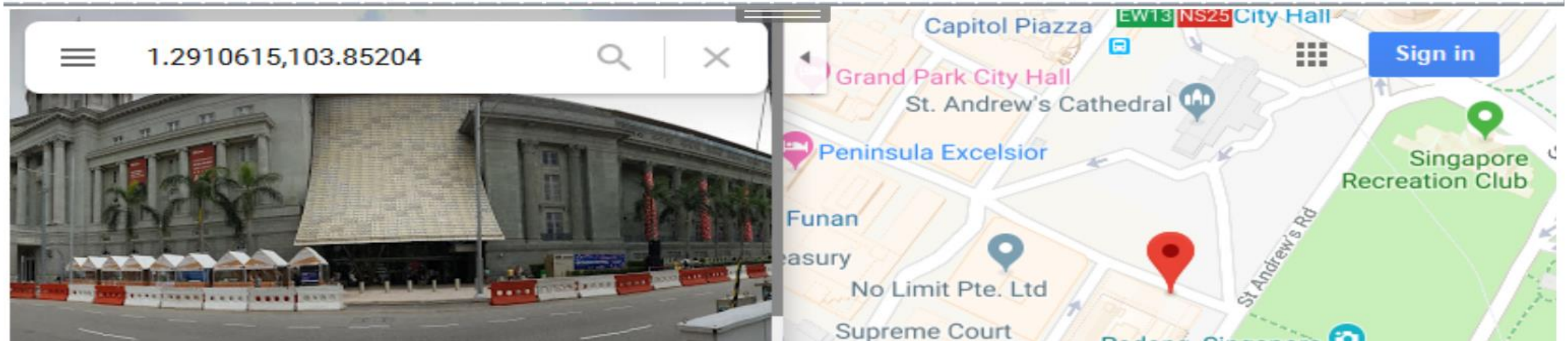
1°17'26.4"N 103°51'06.9"E
1.290660, 103.851910

# Visualization example using OpenStreetMap & Google Maps



When we select a particular circle it is highlighted on the OpenStreetMap and the corresponding location is loaded in the bottom window using Google Maps. In this example, we see that the selected location is an entrance/exit to the National Gallery of Singapore. We get new insights and additional information by combining these 2 maps.

| | |
|---|---|
| Floor: | Level 1 |
| Linegroup: | 1 |
| Time: | 5/12/2018 1:15:41 PM |
| Lat: | 1.2910615 |
| Long: | 103.8520400 |
| Pointorder: | 5,396 |



1.2910615,103.85204

# Control phase and project closeout

- In this phase, we standardise work flows, use control metrics to measure the performance of our algorithms and monitor the improvement.
- We control process performance so that the targets are met and ensure that defects do not occur again.
- We perform a project closeout after ensuring that we have met the client's expectations and requirements successfully.

# Project Deliverables

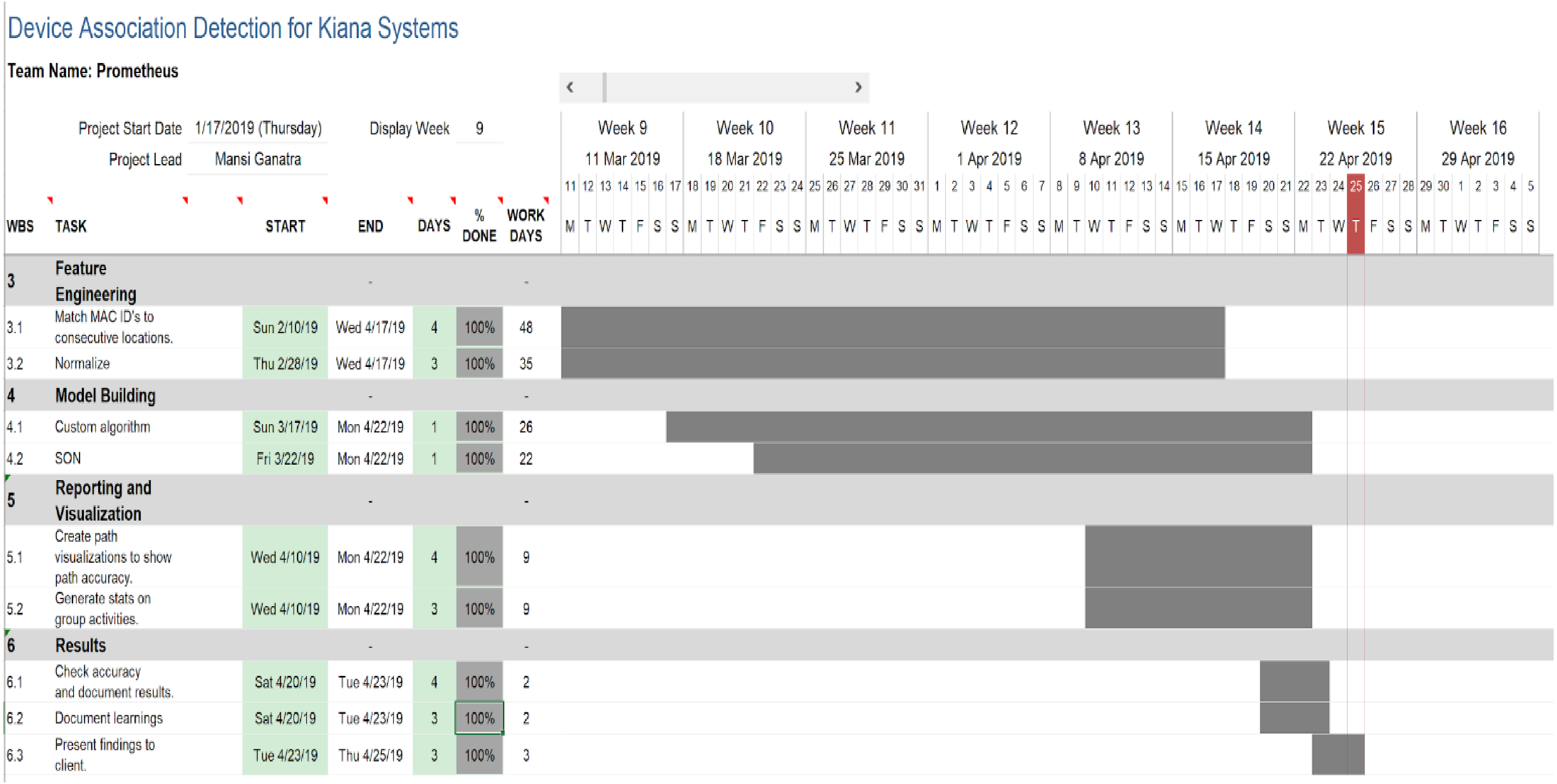
Two Algorithms:

1) Real time Algorithm: On running the code the user will get a list of mac ids  which we believe belong to the same person or maybe they can be a group of people.
2) End of the Day Algorithm: Again it will be a code that a user can run to look at the sample result.
3) Documentation of algorithm.
4) Visualizations of results.

# Process improvement activity for future projects

1) Running the algorithms on the larger datasets with larger computing power.
2) Integrate a user interface with the algorithms.
3) Use a parallelized SON algorithm to improve the end of day algorithm efficiency.
4) Implementing an overlapping time buffer for the realtime algorithm.

# Gantt Chart



## Device Association Detection for Kiana Systems

**Team Name: Prometheus**

Project Start Date 1/17/2019 (Thursday)  Display Week 9

Project Lead Mansi Ganatra

Week 9 11 Mar 2019 | Week 10 18 Mar 2019 | Week 11 25 Mar 2019 | Week 12 1 Apr 2019 | Week 13 8 Apr 2019 | Week 14 15 Apr 2019 | Week 15 22 Apr 2019 | Week 16 29 Apr 2019

| WBS | TASK | START | END | DAYS | % DONE | WORK DAYS |
|---|---|---|---|---|---|---|
| **3** | **Feature Engineering** | | - | | | - |
| 3.1 | Match MAC ID's to consecutive locations. | Sun 2/10/19 | Wed 4/17/19 | 4 | 100% | 48 |
| 3.2 | Normalize | Thu 2/28/19 | Wed 4/17/19 | 3 | 100% | 35 |
| **4** | **Model Building** | | - | | | - |
| 4.1 | Custom algorithm | Sun 3/17/19 | Mon 4/22/19 | 1 | 100% | 26 |
| 4.2 | SON | Fri 3/22/19 | Mon 4/22/19 | 1 | 100% | 22 |
| **5** | **Reporting and Visualization** | | - | | | - |
| 5.1 | Create path visualizations to show path accuracy. | Wed 4/10/19 | Mon 4/22/19 | 4 | 100% | 9 |
| 5.2 | Generate stats on group activities. | Wed 4/10/19 | Mon 4/22/19 | 3 | 100% | 9 |
| **6** | **Results** | | - | | | - |
| 6.1 | Check accuracy and document results. | Sat 4/20/19 | Tue 4/23/19 | 4 | 100% | 2 |
| 6.2 | Document learnings | Sat 4/20/19 | Tue 4/23/19 | 3 | 100% | 2 |
| 6.3 | Present findings to client. | Tue 4/23/19 | Thu 4/25/19 | 3 | 100% | 3 |

# Technical Learnings and Takeaways

- Convert Strings to numbers or hash.
- Start from the smallest sample and build your way up.

# Project Management Learning and Takeaways

- Delivering a project to a client using the Six Sigma principles.
- Time management considering each team member has a different schedule.
- Sticking to the timeline and working effectively as a team.

# References

Heagney, J. (2011). *Fundamentals of Project Management* (4th ed.). New York, NY: The Association.

Laasonen K. (2005) Clustering and Prediction of Mobile User Routes from Cellular Data. In: Jorge A.M., Torgo L., Brazdil P., Camacho R., Gama J. (eds) Knowledge Discovery in Databases: PKDD 2005. PKDD 2005. Lecture Notes in Computer Science, vol 3721. Springer, Berlin, Heidelberg

S. Muthuramalingam and R. Rajaram and Kothai Pethaperumal and V. K. D. Dev (2010). A Dynamic Clustering Algorithm for MANETs by modifying Weighted Clustering Algorithm with Mobility Prediction

Youngjun Son, Mokdong Chung. (2014). Digital Forensics for Android Location Information using Hierarchical Clustering. Journal of the Institute of Electronics and Information Engineers, 51(6), 143-151.

http://robowiki.net/wiki/Dynamic_Clustering