User Location Clustering

Neal Patel

CLASSPASS

Agenda

Background

- Hyper-local industry
- Challenges

Distance Estimation

- KNN – modified (SQL)
- Results

Location Prediction

- DBSCAN Clustering Algorithm Overview
- Results
- Validation

Next Steps

CLASSPASS

Background

Fitness is a Hyper Local Industry

COST | CONVENIENCE | CALORIES

Location Based Context is Critical Information

- Class Recommendations – *User Engagement*
- Studio Optimizations – *Studio Engagement*
- Inventory Management – *Balancing Supply & Demand*
- User-level preference / availablility: *Churn Prediction*

Primary Challenges

- Personal information for users is *private*
- We do not have any labeled data for validation

Solution: 2-pronged approach to validating results

- Estimate willingness to travel using reservation data
- Predict locations and measure relative distances to reservations

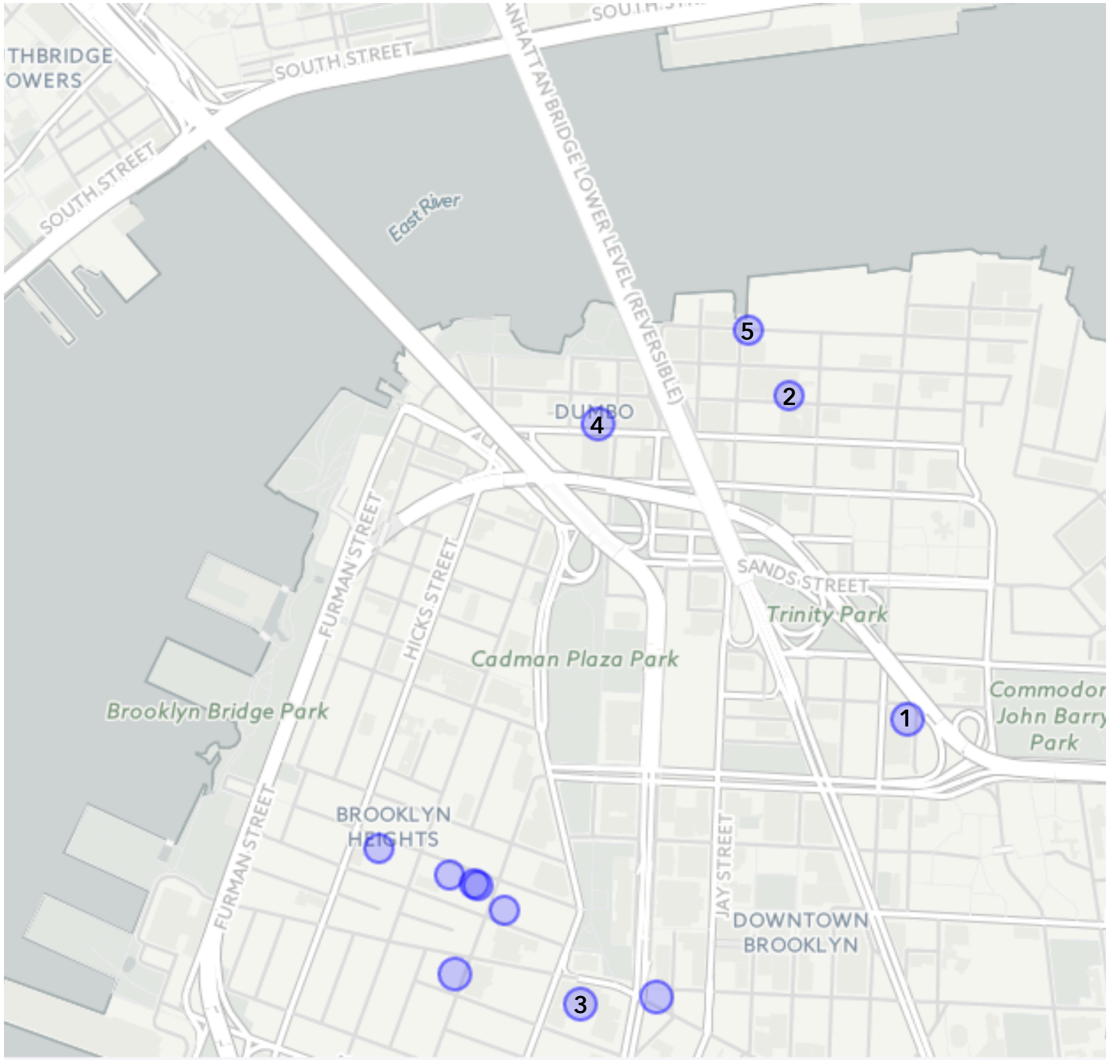
Distance Estimation

How Far Are Users Willing to Travel for Class?

Personal Information is protected, we must build an understanding without it

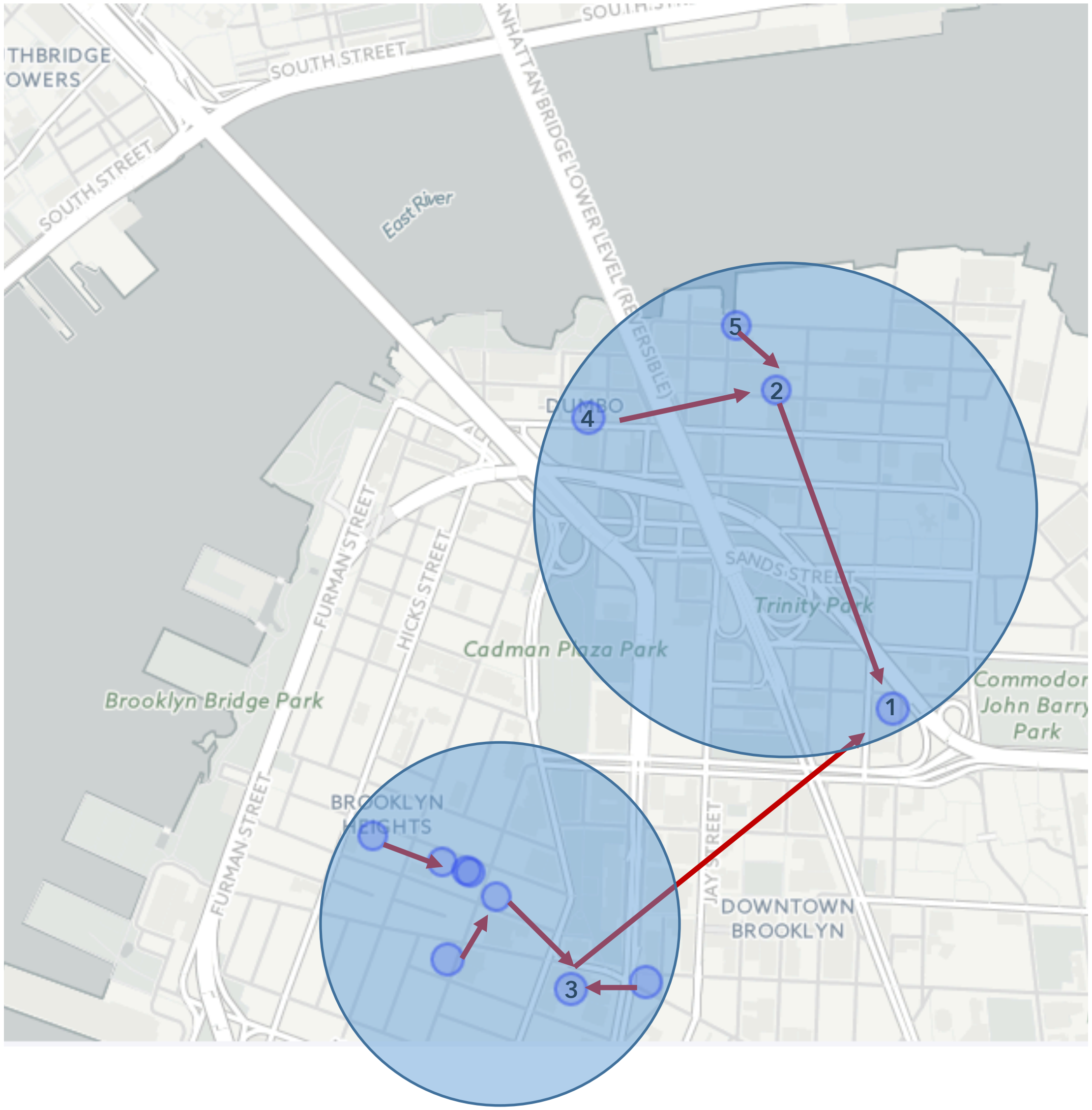
- Data:
  - *Reservation location (latitude / longitude)*
  - *Reservation time*
  - *Reservation frequency*

K – Nearest Neighbors: Modified (SQL)



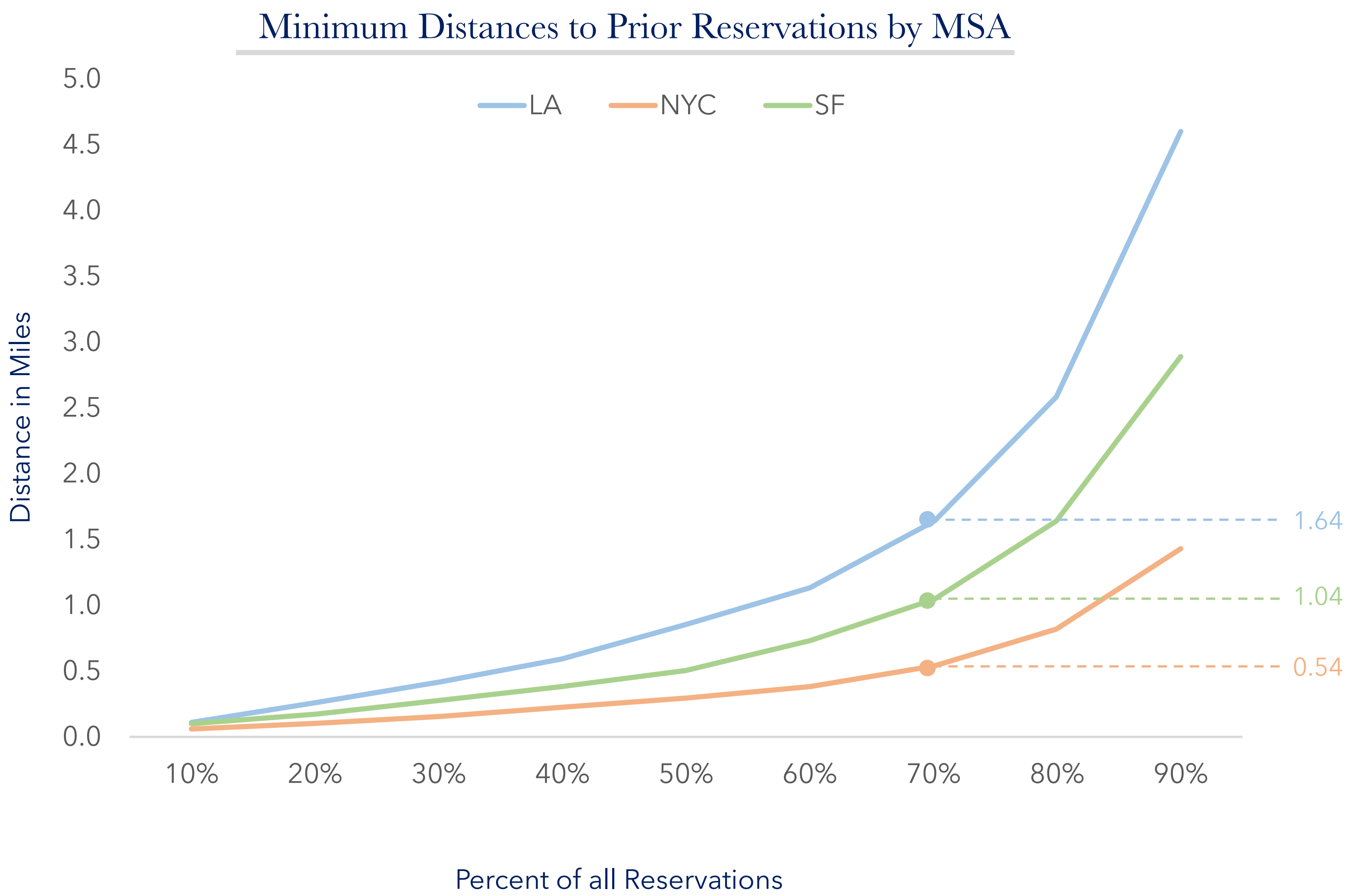
- Users behaviors tend to cluster near "anchor points"
- Using a method inspired by KNN, we can produce an estimate
- We are essentially trying to classify a given reservation as being part of a "cluster"
- However, we are not interested in classifications, rather, the distances between the points in a cluster

K – Nearest Neighbors: Modified (SQL)



- Look at every incremental reservation and identify the *nearest* previously attended venue
- Log the minimum distances and produce a histogram
- Use a 'cutoff' to remove long-tail and use value @ threshold as the estimate
- This will likely 'underestimate' the distance

K – Nearest Neighbors: Modified (SQL)



Location Prediction

CLASSPASS

User Searches Provide Location Data

- Majority of our users make use of "Search Near Me" feature
- Enables logging of GPS data
- *Assumption:* Users will most often be browsing for classes at an "anchor point" (i.e. Home, Work, Sig. Other, etc.)

DBSCAN Algorithm - Overview

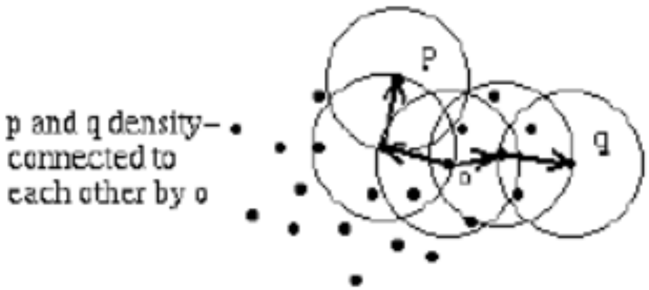


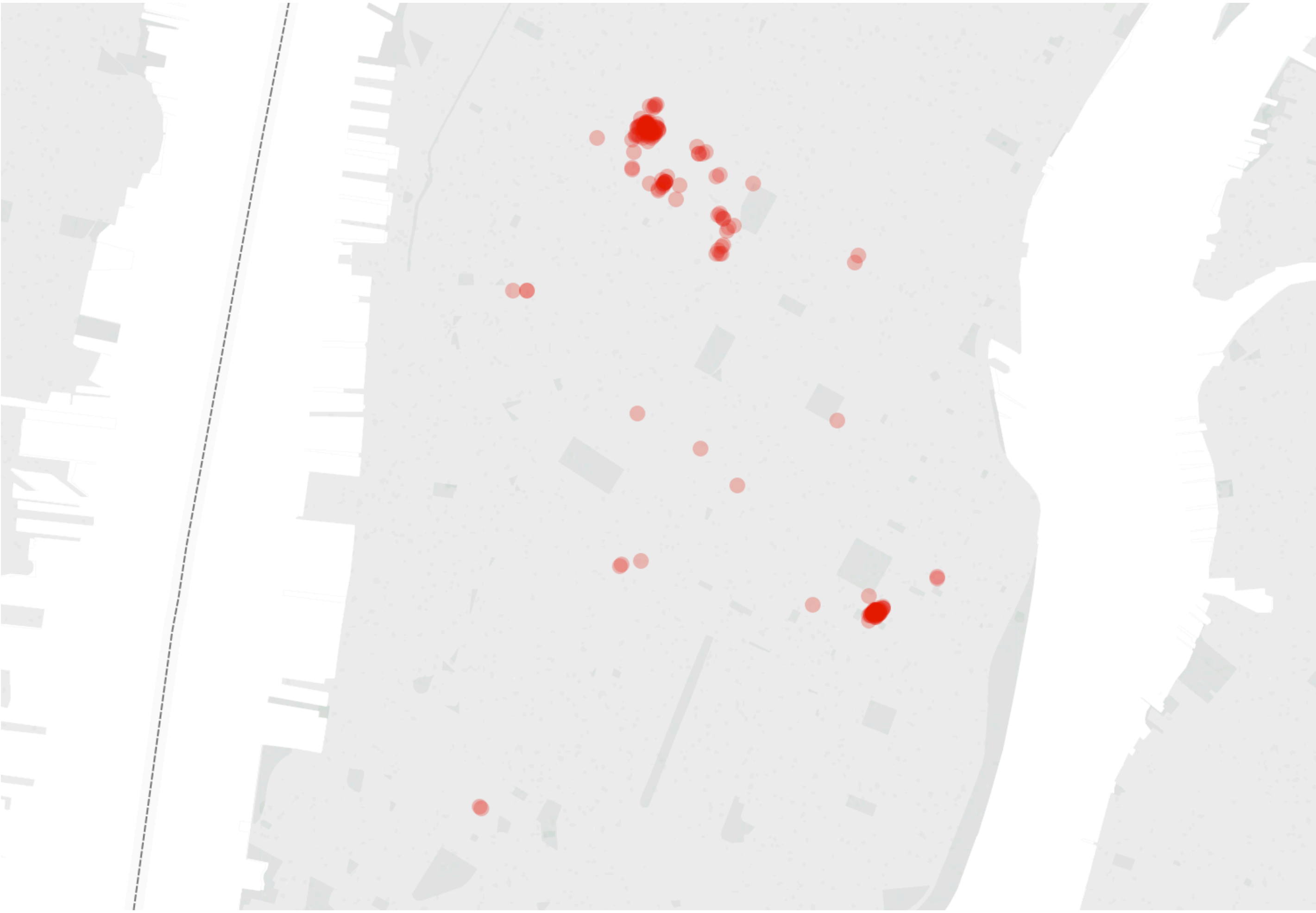
Figure 5. Density connectivity.

- User coordinates from mobile phones are highly variable
  - Creates lots of 'noise'

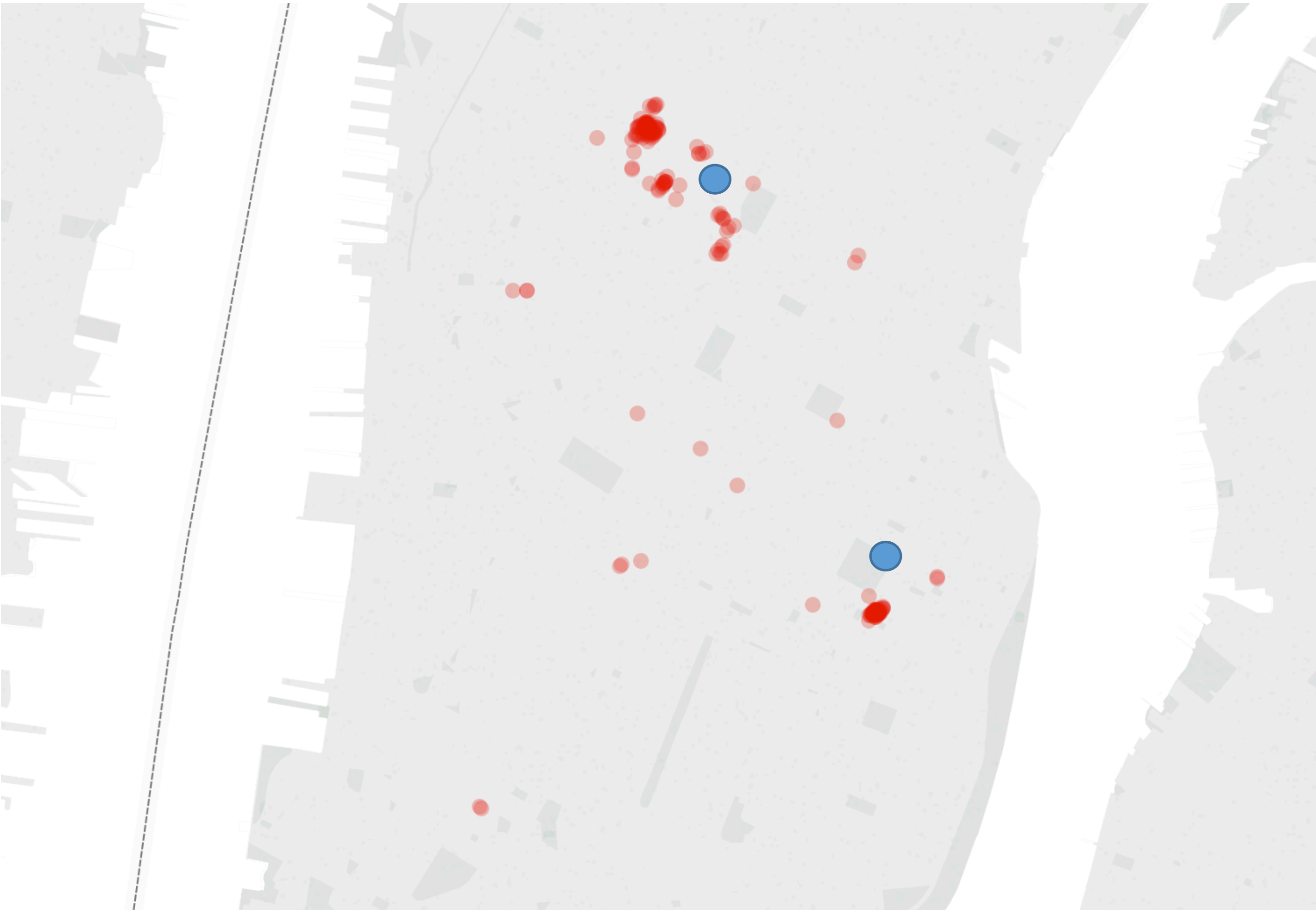
- DBSCAN identifies points that are connected by a relative "density" value – eliminates noise points

- Goal: Identify clusters, take center-point as "prediction"

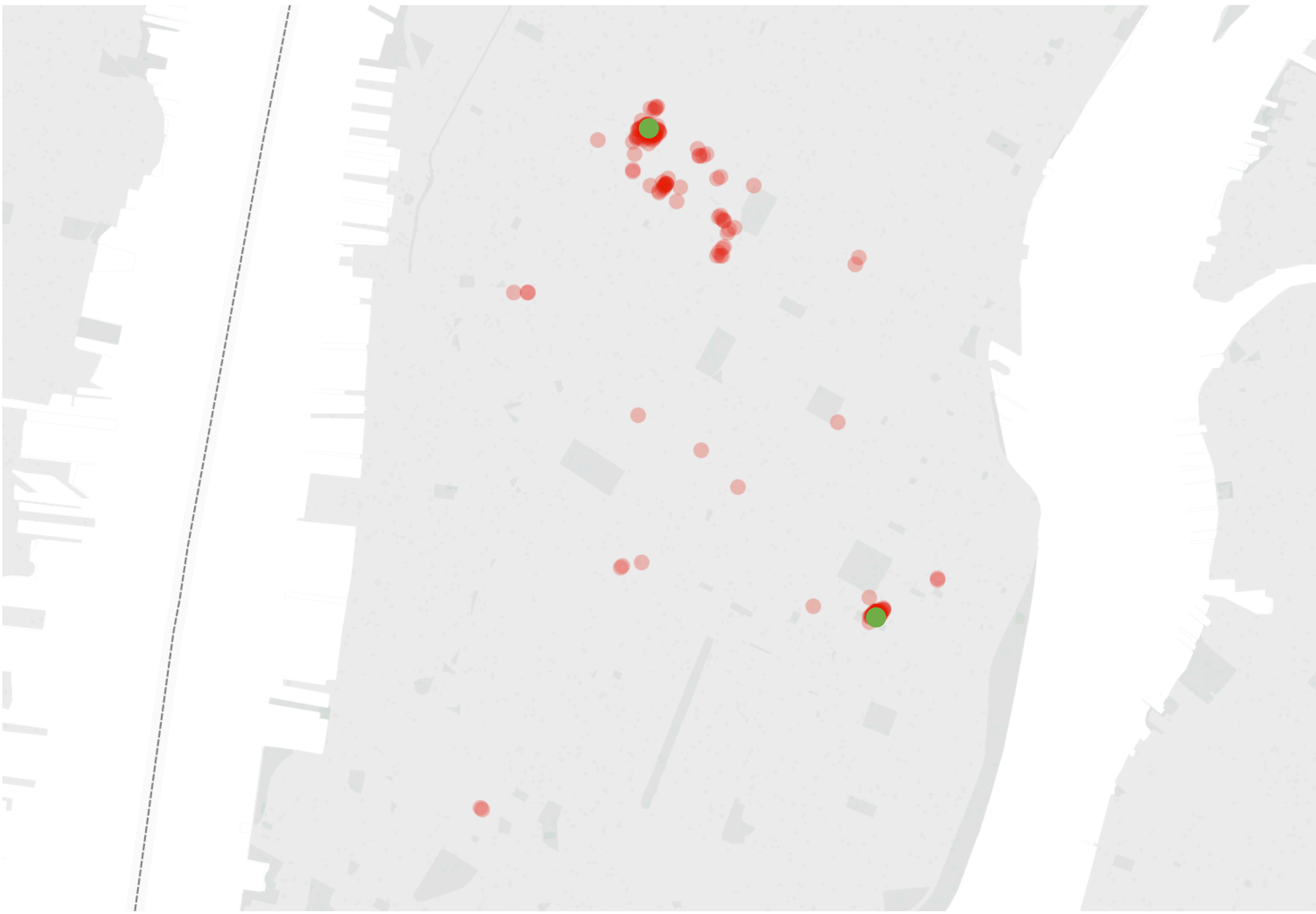
DBSCAN Algorithm – Sample Data



DBSCAN Algorithm – K-Mean Comparison

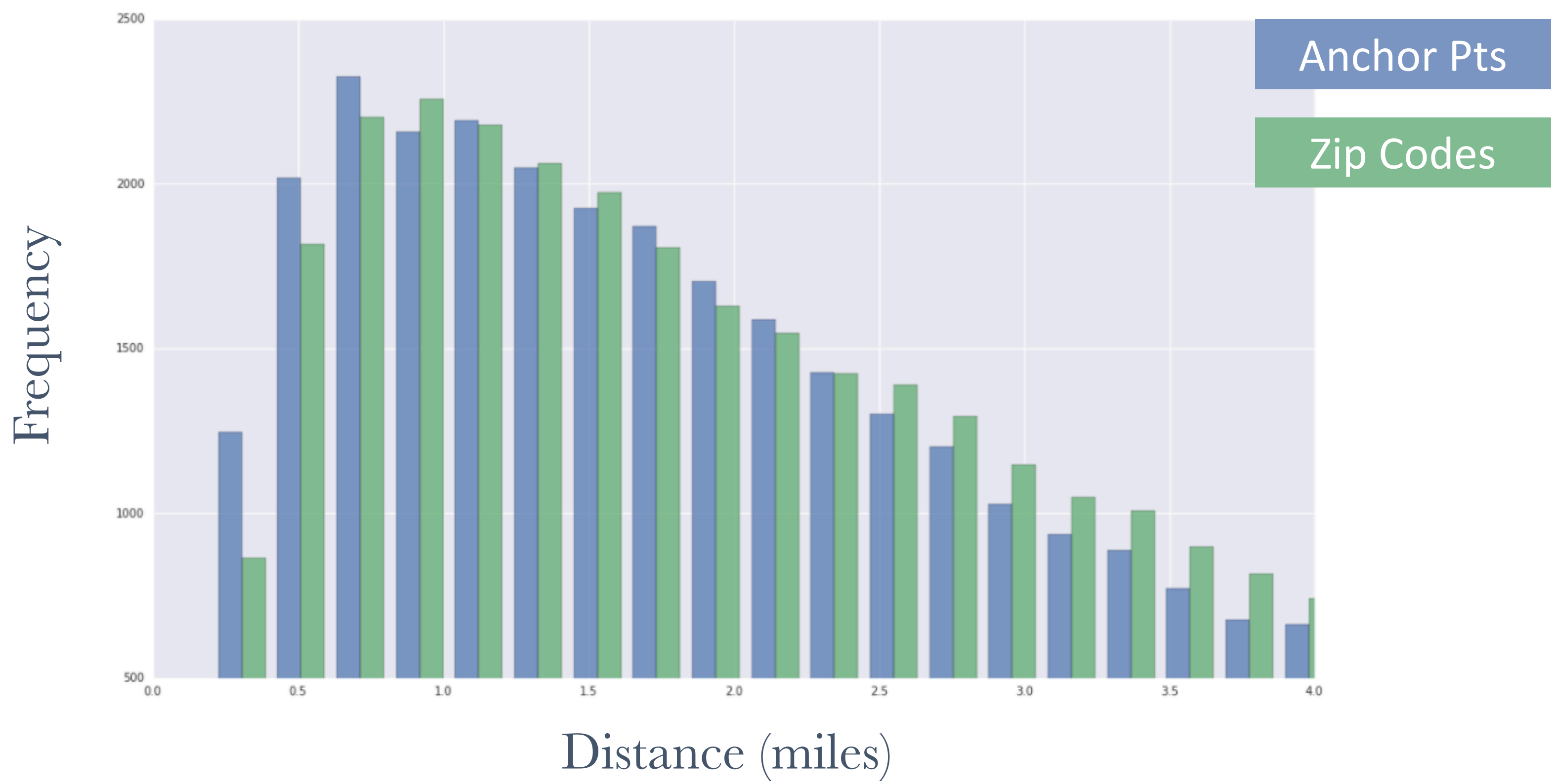


DBSCAN – Location Predictions



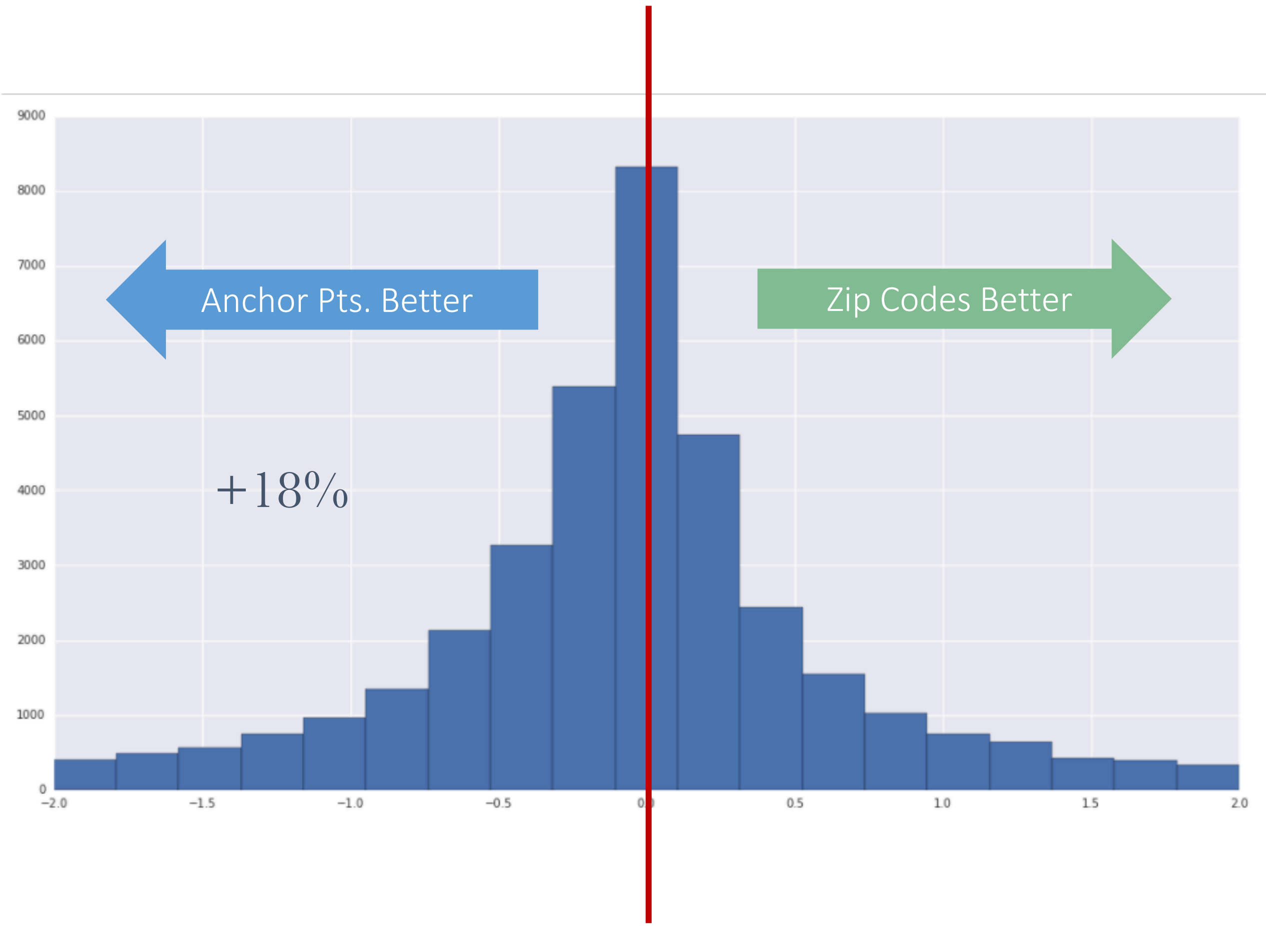
Validation: Zip vs. Anchors

- Users provide Zip Codes when signing-up
- We can compare performance of Zip Codes vs. Predicted Anchor Points
  - Performance measured on relative distances to user reservations



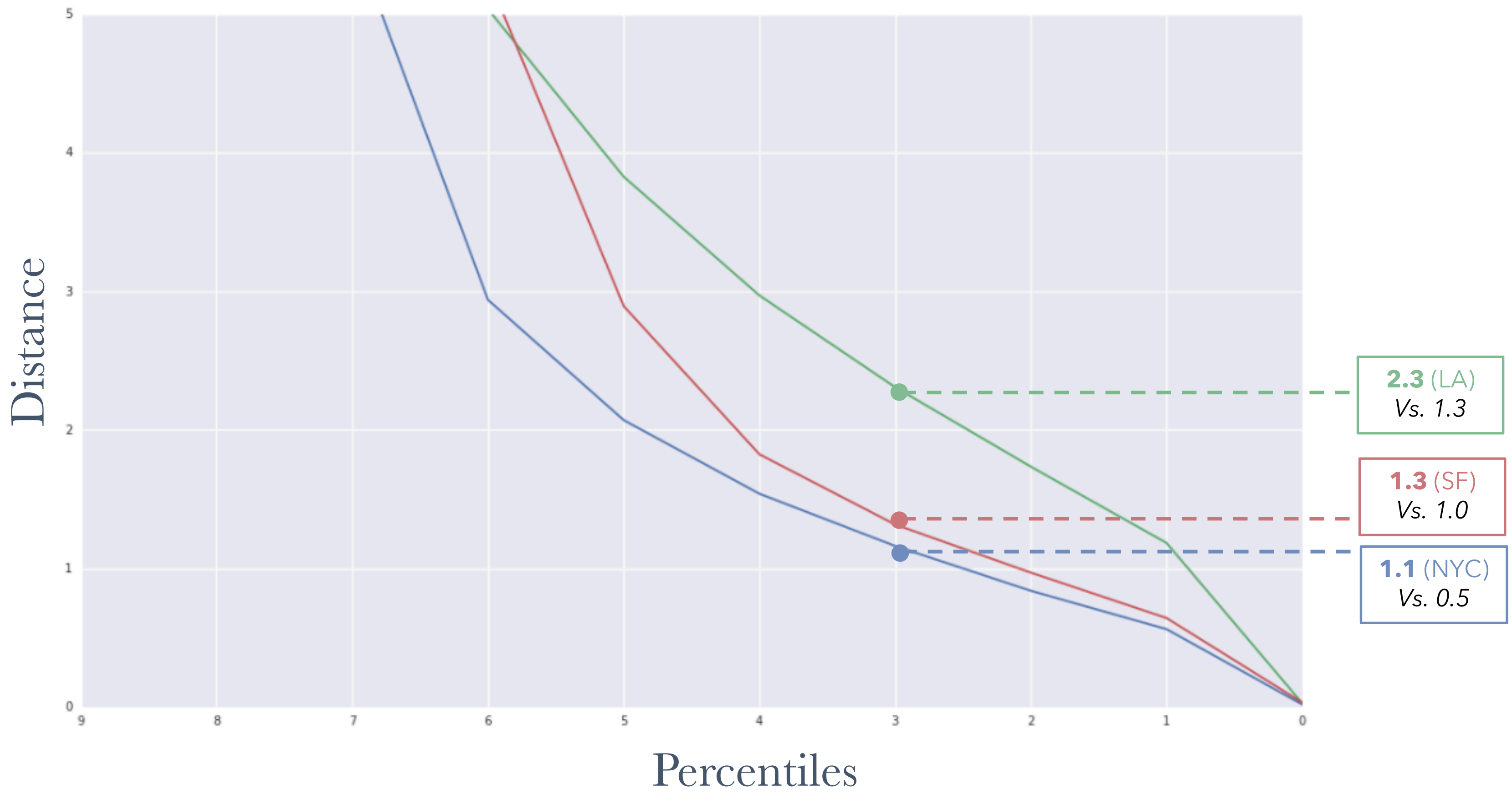
Validation: Zip vs. Anchors

- Distribution of the deltas(d) ($d = Anchor\ Distance - Zip\ Distance$) provides additional insight



Validation: Zip vs. Anchors

- Distribution of the deltas(d) ($d = Anchor\ Distance - Zip\ Distance$) provides additional insight



Next Steps

The Road Ahead

1. Tweak algorithm further to improve performance beyond 16%

2. Apply model to full user database

3. Develop understanding of user-level inventory trends

4. Explore models to use above inputs as predictors of user churn:
    - Logistic Regression
    - Classification Problem