

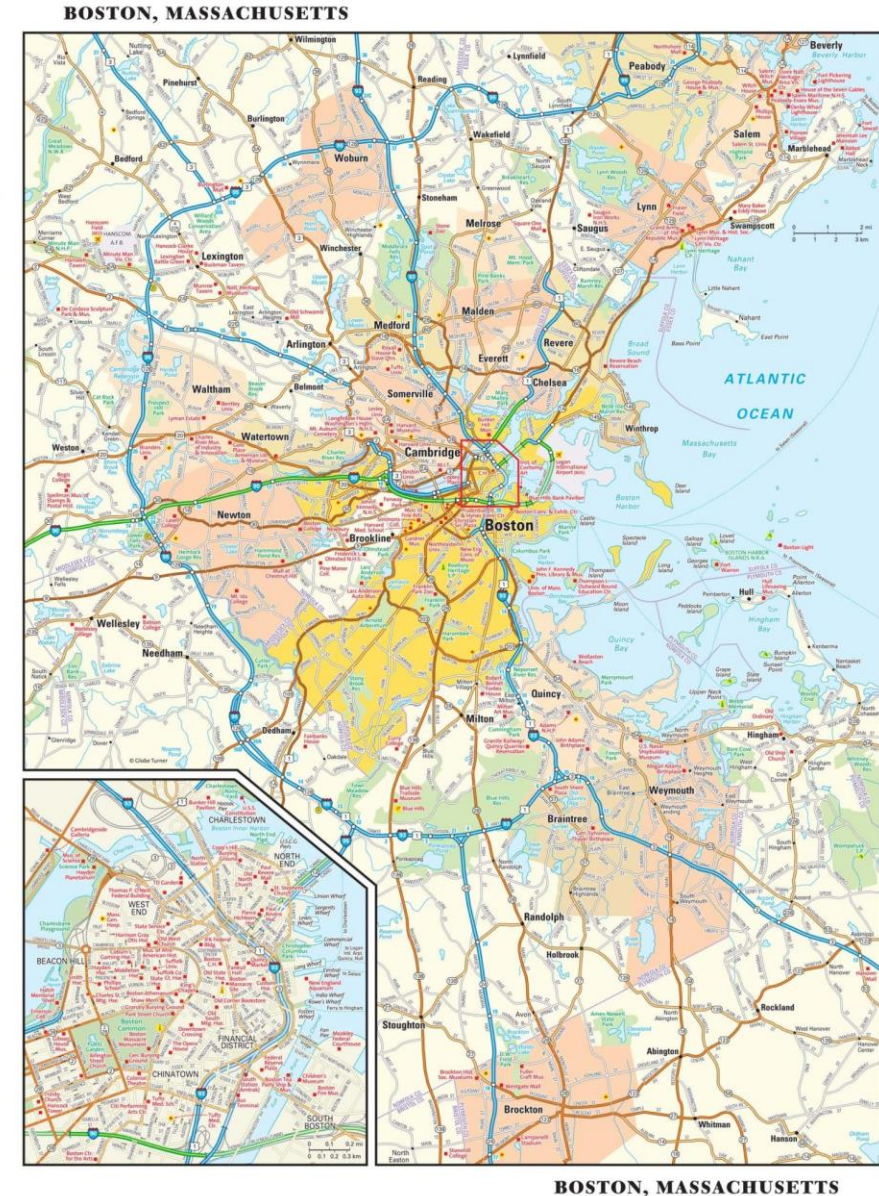
# Coursera IBM Capstone

## Exploring the Data Locality of Restaurants in Boston

Kody Richardson

# Introduction

- We are searching for the best place to build a new Chinese Restaurant in Boston
- We are looking for a place that has a golden ratio of Chinese Restaurant density to restaurant density of other types of food.



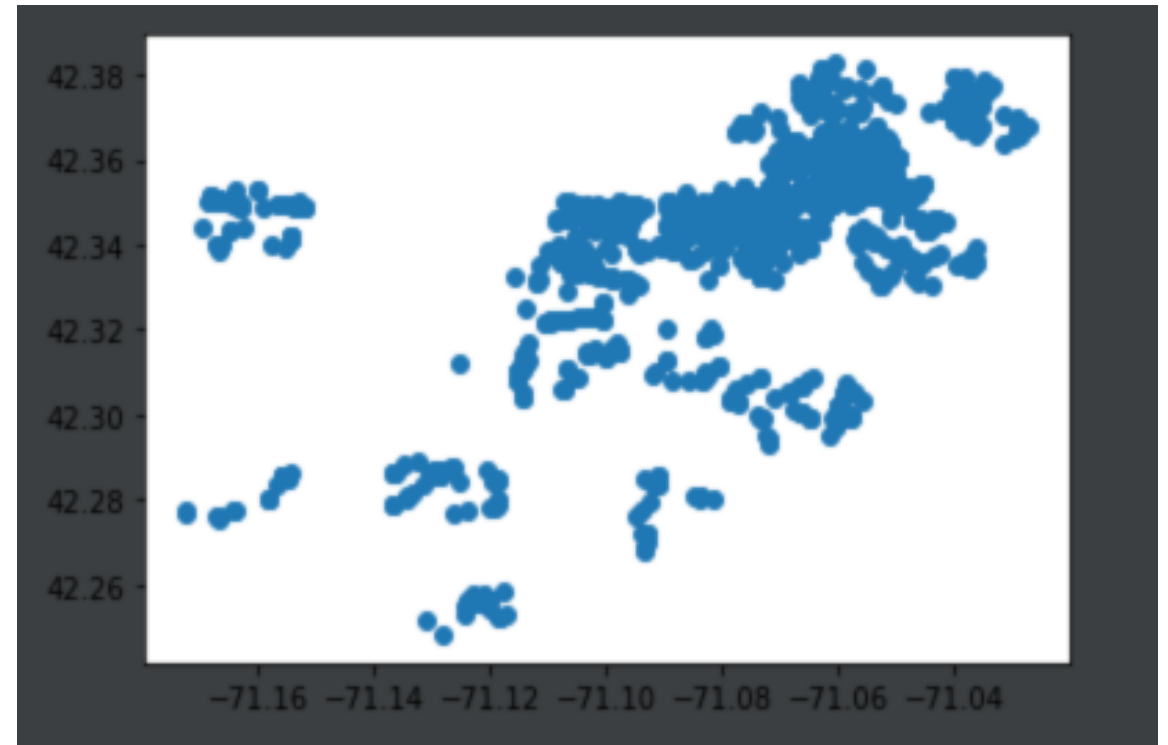
# Data

- We will be pulling two types of data.
- We will use the location of Boston Districts from an external CSV
- We will use foursquare to gather all the restaurants and their types of food in Boston.

	Neighborhood	Latitude	Longitude
0	Allson	42.352900	71.132100
1	Back Bay	42.351294	-71.080356
2	Bay Village	42.349100	-71.068000
3	Beacon Hill	42.358300	-71.066100
4	Brighton	42.346400	-71.162700

# Data Analysis

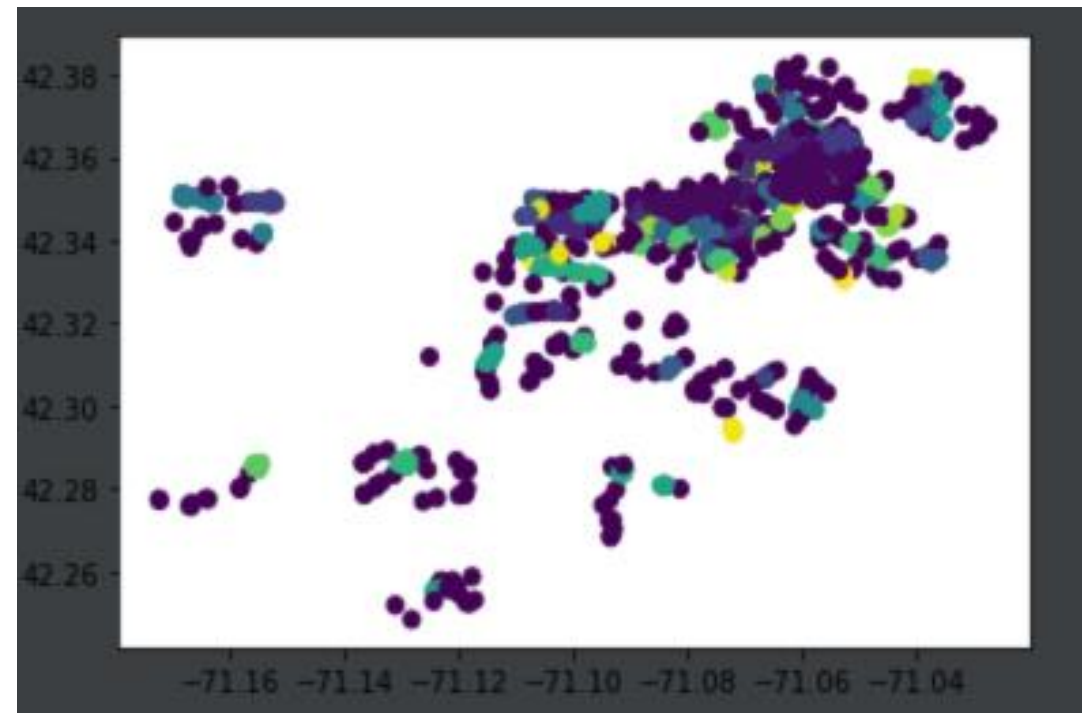
- We pulled all the data from foursquare on each restaurant type. The total amount of data pulled equated to 57,200 restaurants.
- The graph plots each restaurant based on the longitude and latitude.



# Cluster Model

- We generated our clusters of restaurants using DB scan which found us 92 clusters.
- The code to the right was used to cluster the restaurants.
- The graph shows the location of our clusters on top of the groupings of restaurants

```
# Epsilon dictates how tight the density groups can be.  
# Min samples says how few data points are needed for a cluster to form  
# Cluster list is a numpy array containing the X and Y coordinates of every Venue  
dbScan = DBSCAN(eps=0.001, min_samples=5).fit_predict(clusterList)  clusterList:
```



# Results

- We ran the code to the right to figure out the best possible longitude and latitude for a new Chinese restaurant.
- The given general location was:
  - Longitude: -71.05890045602365
  - Latitude: 42.35555054151881

```
for idx in range(len(clusterList)): clusterList: <class 'list': ['78', '59', '3
    weight = numberOfResterauntsPerCluster[idx]/totalVenues numberOfResterauntsP
    percentX = numberOfChinesePerCluster[idx]/numberOfResterauntsPerCluster[idx]
    value = weight*percentX weight: 0.010632911392405063 percentX: 0.0
    if percentX !=0: percentX: 0.0
        finalResults.append( (clusterList[idx],value) ) finalResults: <class 'Li
```

```
for idx in range(len(clusterResultList)): clusterR
    if clusterResultList[idx] == maxName: clusterR
        counter+=1 counter: 0
        longCounter = longCounter + longList[idx]
        latCounter = latCounter + latList[idx] l
```

# Conclusion

- While we were able to find a general location for our new restaurant. We found that we could make our algorithm more precise by including different data.
- We could include data on population density of each district, and the average household income of each family in the districts.
- This would allow us to bring population into the factors as well as to see what kind of Chinese Restaurant we would want to build.
- There is plenty more work to be done and I look forward to exploring more of this project.