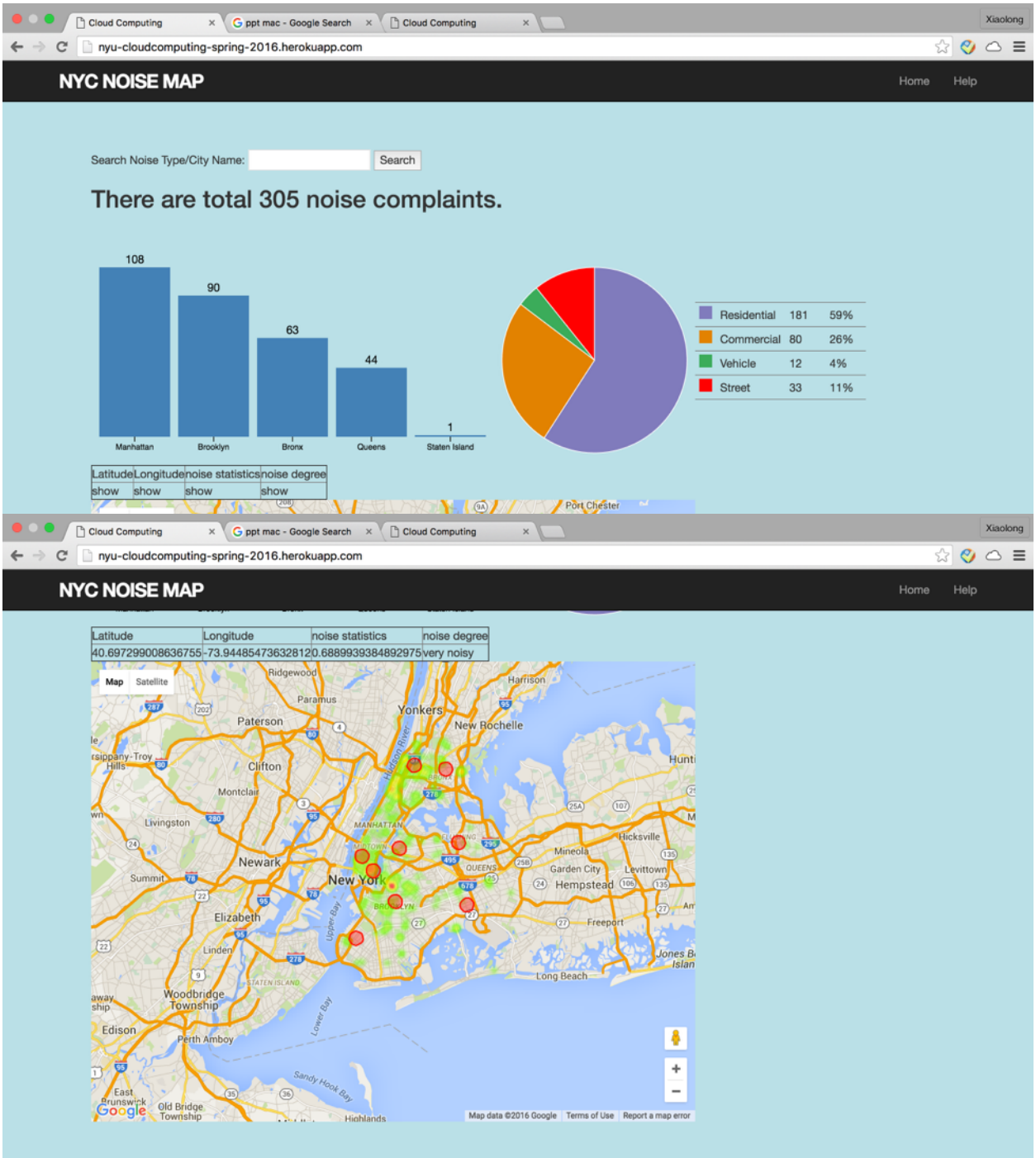
Cloud Computing Final Report
NYC Noise map

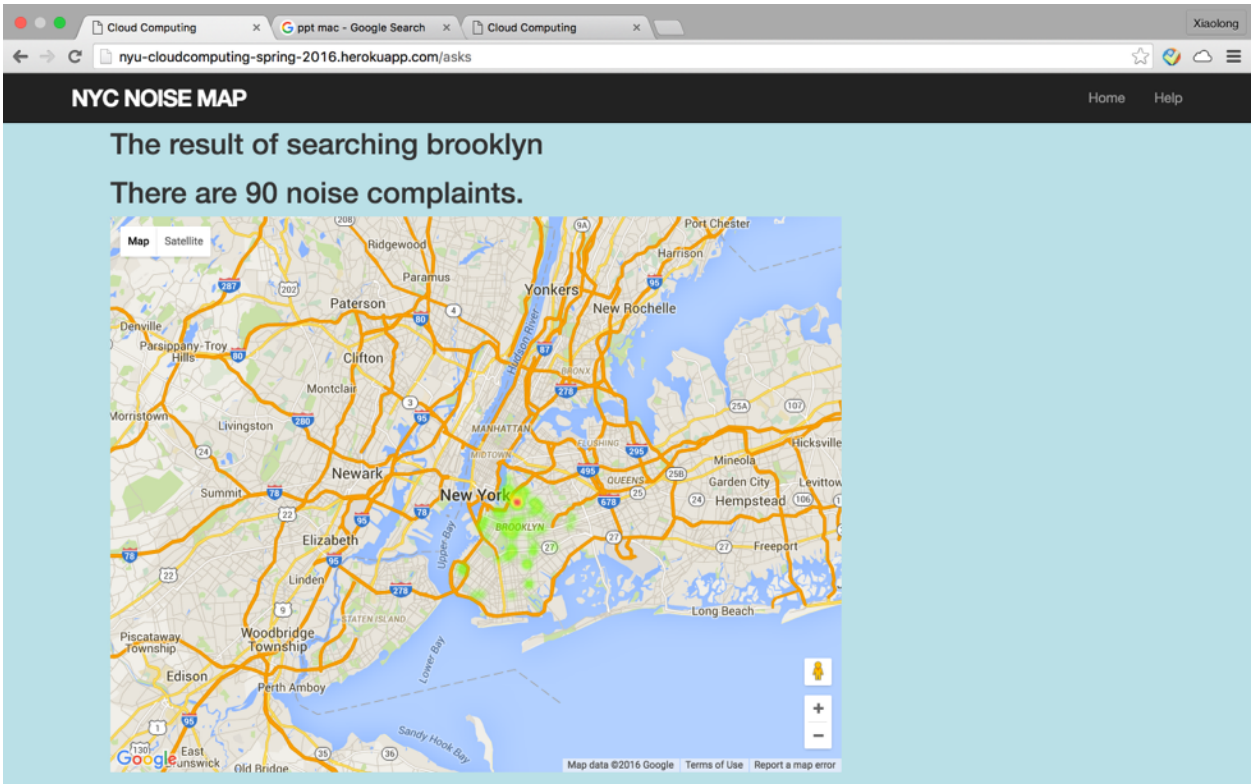
Xiaolong Jiang, Guoqing Xiong

We've build a application of gathering and analysis NYC 311 Service Request Data. We choose noise complaint to analysis and visualize on the Google map and D3 library's pie/bar chart, which shows user the noise distribution in format of heat map, and statistic result of different



complaints. Moreover, we can search based on the City name or street name, or type of complaints.
There are four major types of complaints as we can see in the pie/bar chart — they are residential, vehicle, street and commercial. NYC city are Manhattan, Brooklyn, Queens, Bronx and Staten Island.

We stored the NYC open data, and upload it onto CloudSearch to build index. It can be searched based on Complaint **Type, City, Street, and description of complaint**. We use HTTP endpoint request to fetch Json format data with ruby on rails 'net/http' and 'json' gem. As we can see in the figure below, it is the result of search 'Brooklyn'.



As I mentioned above, we use D3.js library and google map to visualize the data. In the D3 library, we use a pie chart and a bar chart together to demonstrate distribution of complaint types in cities, and each complaint's

distribution in these 5 cities. For example, (introduce bar chart and pie chart). In the Google map, we are using heat map to demonstrate noisy scale. The red zone represents that there are many complaints at that region. Meanwhile, click on the google map will show the noise degree of that location. It ranges from very quiet to very noisy, which is based on the K-means machine learning algorithm result.

We use spark to analysis data. We implement a K-means machine learning algorithm to cluster the geoLocation points into 10 different regions, which represent a region's center that contains a lot complaints. The noise scale of a single location is calculated based on distance from these 10 different region centers. Here is the spark algorithm and the result we get from K-means machine learning.

```
In [5]: from pyspark.mllib.clustering import KMeans, KMeansModel
        from numpy import array
        from math import sqrt

        data = sc.textFile("point1.txt")
        parsedData = data.map(lambda line: array([float(x) for x in line.split("\t")]))
        clusters = KMeans.train(parsedData, 10, maxIterations=5,
                runs=3, initializationMode="random")
        def error(point):
            center = clusters.centers[clusters.predict(point)]
            return sqrt(sum([x**2 for x in (point - center)]))

            filename = "output/kmeansCentersCSV2.csv"
            logfile = open(filename, "w")
            i = 0
            while i < len(hehe):
        ----line = hehe[i]
        ----writestring = "%s,%s\n" % (str(line[0]), str(line[1]))
        ----logfile.write(writestring)
        ----i += 1
            logfile.close()
            clusterCenters=clusters.clusterCenters
            clusterCenters

Out[5]: [array([ 40.76564313, -73.95405624]),
         array([ 40.82492466, -73.93701465]),
         array([ 40.8448316, -73.8742622]),
         array([ 40.86108478, -73.91168021]),
         array([ 40.67745568, -73.90666845]),
         array([ 40.71152867, -73.97072419]),
         array([ 40.63197806, -73.98262497]),
         array([ 40.88065355, -73.85916471]),
         array([ 40.85205434, -73.82903921]),
         array([ 40.72381085, -73.81731067])]
```

Cloud Computing Tools

1. S3: To store open source data, and result of spark machine learning.
2. CloudSearch: Index data, search based on key word and specific field.
3. Spark: Implement K-means machine learning algorithm to cluster locations.
4. Heroku: Deploy Ruby on Rails Backend.

Procedure of NYC Noise Map