Data Science at Scale – Capstone project

author: Takayuki Kaisen
address: ksn0215(at)gmail.com
Date: May 19, 2018
Version: 1.0

0. Notice

I uploaded my code in my Github page: https://github.com/ksnt/Predictor-of-blights-in-Detroit
I am sorry for messy format of my notebook and report. I will do refactoring in the future. Therefore, these might not be available in the future on the web.

1. Introduction

1.1. Background

Detroit flourished in the motor industry, however the city has gone to pot. If you walk down the street in Detroit on Google map, you could find many collapsed buildings and a defunct city. If you want to know how and why Detroit has been desolated, you could find many videos in YouTube and some articles on the web. For planners or developers of the city, in this situation, finding and predicting where will be blighted in the future is a big challenge and worthwhile from the view of city's health.

1.2. Goal of this project

In this project, through data analysis, we will be building models for classification and prediction of blights in Detroit using Machine Learning techniques. We used data including crime occurrences, 311 calls, demolition permits, and issued information of blights.

1.3. Main result in this project

Here, Logistic Regression, Boosting, and Random Forest are used for classification and prediction. Features are the number of blights, the number of 311 calls, and the number of crimes in a region. In conclusion, Boosting model using all three features has the best outcome: the accuracy to test data is 90.08 %.

2. Data, Method, and Result

Overview of the data is below:

File Name	Description
detroit-311.csv	The data of 311 calls in Detroit, 2016.
detroit-blight-violations.csv	The data of blight violations in Detroit, 2016. The City has ordinances that address how property owners must maintain the exterior of their property. A blight violation is issued when an owner fails to follow these ordinances.
detroit-crime.csv	The data of crimes in Detroit, 2016.
detroit-demolition-permits.tsv	The data of demolition permits in Detroit, 2016

2.1. Data Exploration

Crime data

In the crime data, you find many points out of Detroit. You can confirm those points in the left figure below.

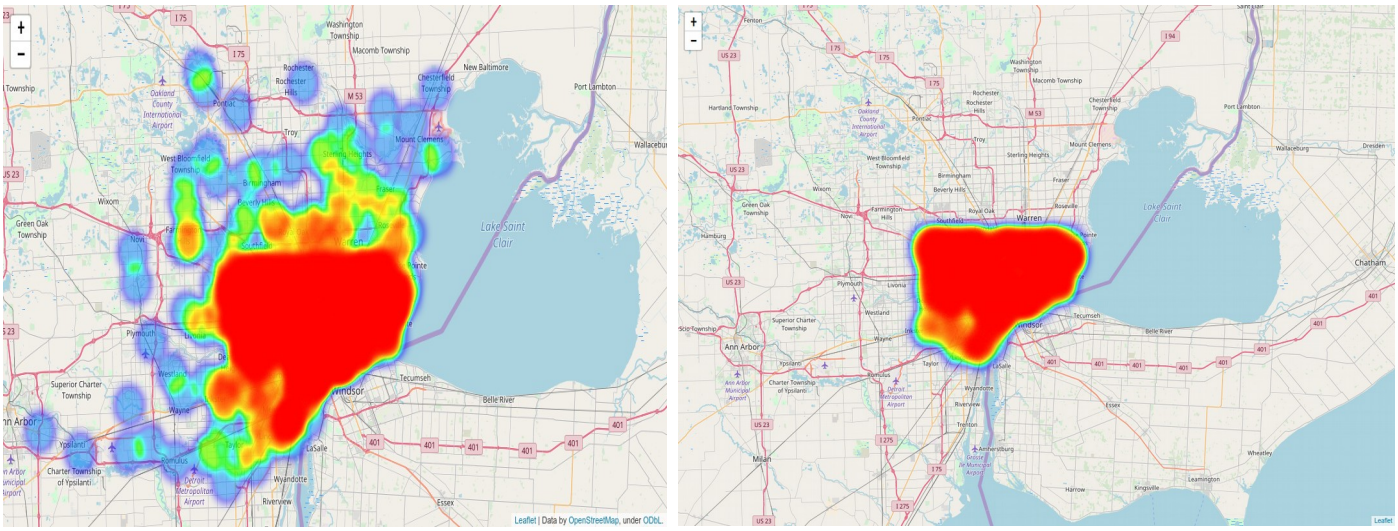


Figure1 Maps before removing outliers and after removing outliers

To remove those data, I decided boundaries of Detroit like this:

```
north_bound = 42.443509 # East 8 Mile Road
south_bound = 42.255240 # Boynton
east_bound  = -82.910255  # Grosse Pointe Farmes
west_bound  = -83.289626  # Southfield
```

Then, I applied these boundaries to the data:

```
crime_data = crime_data[(crime_data.LAT >= south_bound) & (crime_data.LAT <= north_bound)
& (crime_data.LON <= east_bound) & (crime_data.LON >= west_bound)]
```

The result is the right figure above. Regarding other data, data out of Detroit are not in the file, therefore, I used those data as they are.

2.2. Defining “Buildings”

Here, a way to cluster all incidents that occurred at the same address in this project is described.

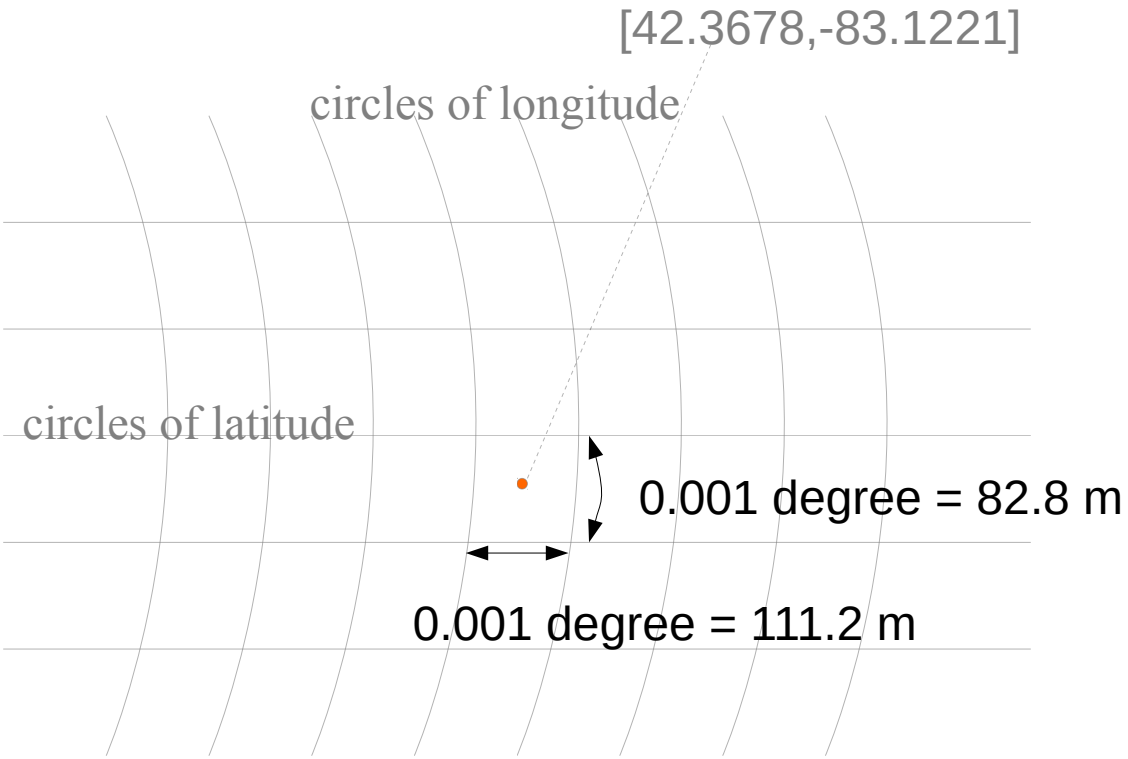


Figure2 Division of map

Approach

In this project, I employ an approach that I convert given 4-digits latitude and longitude data to 3-digits ones. After this operation, the map of Detroit is divided into 0.001 degree by 0.001 degree squares.

Here, I calculate errors after this operation. For this goal, at first, I calculate how to convert degree to second of latitude and longitude and also the length of them presented in degree and second to meter.

(1) On latitude

It is known that radius of the earth is 6378150 meter. Here, let the radius be *r*. Assumed that the earth is round, the equation below holds, where π is the ratio of the circumference of a circle to its diameter:

$2 * \pi * r = 40054782$ (meter)

Thus, 1 degree is calculated below:

40054782 / 360 = 111263.283(meter)

Therefore, 1 second is:

40054782 / (360 * 60 * 60) = 30.9064(meter)

Also, 1 second can be converted to approximately 0.000277777 degree. On the other hand, 1 degree can be converted to 3600 second.

(2) On longitude

The equation below holds under the assumption that the longitude of Detroit is 42 degree.

1 second = 6378150 * cos(42/(180*pi) * 2 * pi) / (360 * 60 * 60)
= 22.9796
≒ 23 meter

Next, I calculate errors when using 2 decimal places and 3 decimal places.

(a) Error when using 2 decimal places (Assume the latitude for Detroit is 42.3678 and the longitude for Detroit is -83.1221)

In this situation, the latitude is represented as 42.36, therefore, any points which value is from 42.3600 to 42.3699 can be seen as the same one point. Thus, observational error is 0.0099degree. It is approximately the same as 0.01degree, and 0.01degree is converted to 36 second. All in all, error can be calculated as follows:

36（second) * 30.9 (meter) ≒ 1200 (meter)

Therefore, this approximation causes 1200 meter error in the direction of latitude. This is too big to think it is the size of one building.

(b) Error when using 3 decimal places (Assume the latitude for Detroit is 42.3678 and the longitude for Detroit is -83.1221)

In this situation, the latitude is represented as 42.367, therefore, any points which value is from 42.3670 to 42.3679 can be seen as the same one point. Thus, observational error is 0.0009degree. It is approximately the same as 0.001degree, and 0.001degree is converted to 3.6 second. All in all, error can be calculated as follows:

3.6（second) * 30.9 (meter) ≒ 120 (meter)

Therefore, this approximation causes 120 meter error in the direction of latitude. This is proper size as the size of one building.

In the same way, error in the direction of longitude can be calculated. When using 3 decimal places regarding longitude, longitude is represented as -83.122. Any points which value is from -83.1220 to -83.1229 can be seen as the same one point. Thus, observational error for longitude is 0.0009 degree as well as latitude. It is approximately the same as 0.001degree, and 0.001degree is converted to 3.6 second. All in all, error can be calculated as follows:

3.6（second) * 23 (meter) ≒ 82.8 (meter)

Therefore, this approximation causes 82.8 meter error in the direction of latitude. This is proper size as the size of one building.

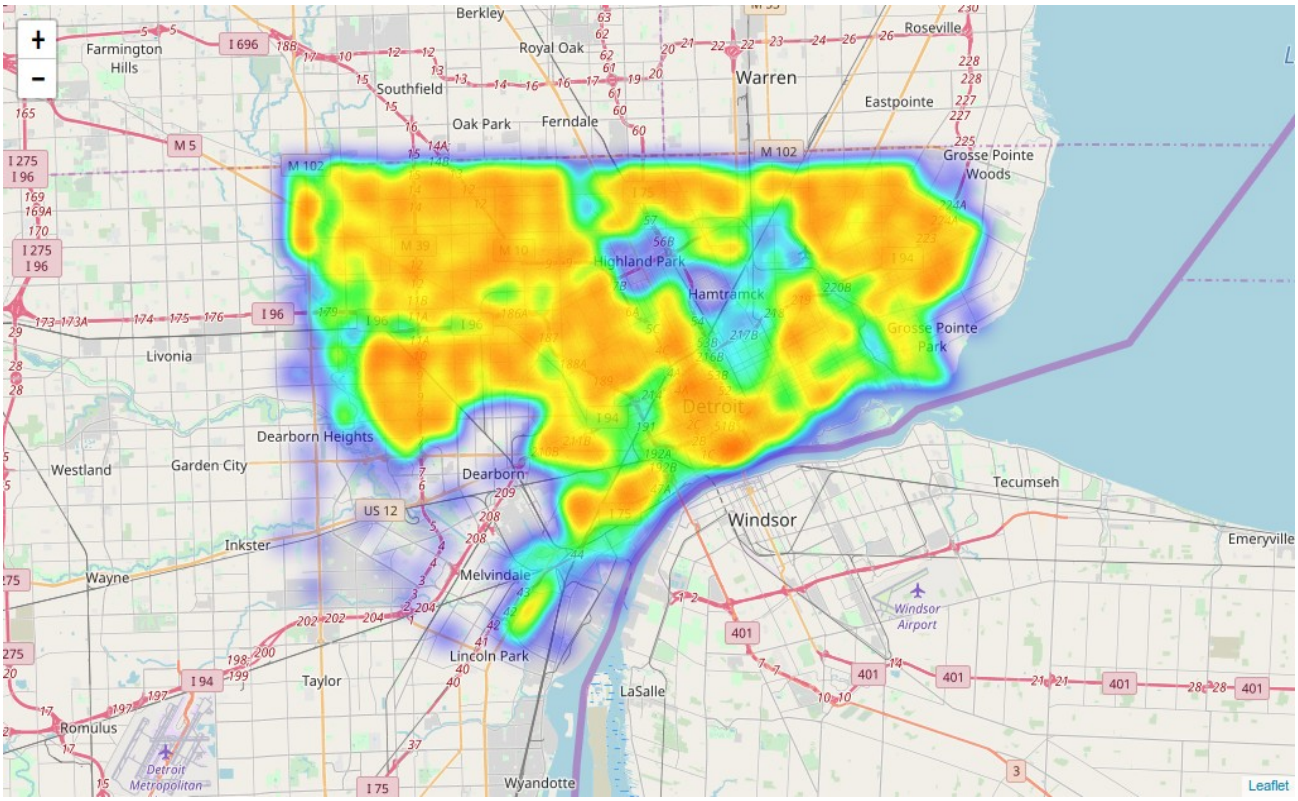
Consequently, when using 3 decimal places, resolution can be lower but it is useful for clustering. Using 3 decimal places rather than 4 decimal places, any points in the 82.8 meter by 111.24 meter square can not be identified. I make use of this property by seeing the 82.8 meter by 111.24 meter square as a building. (See Figure2)

Besides, after this operation, I create ids such as ’building_id’ encoding pairs of latitude and longitude and give them to each building.

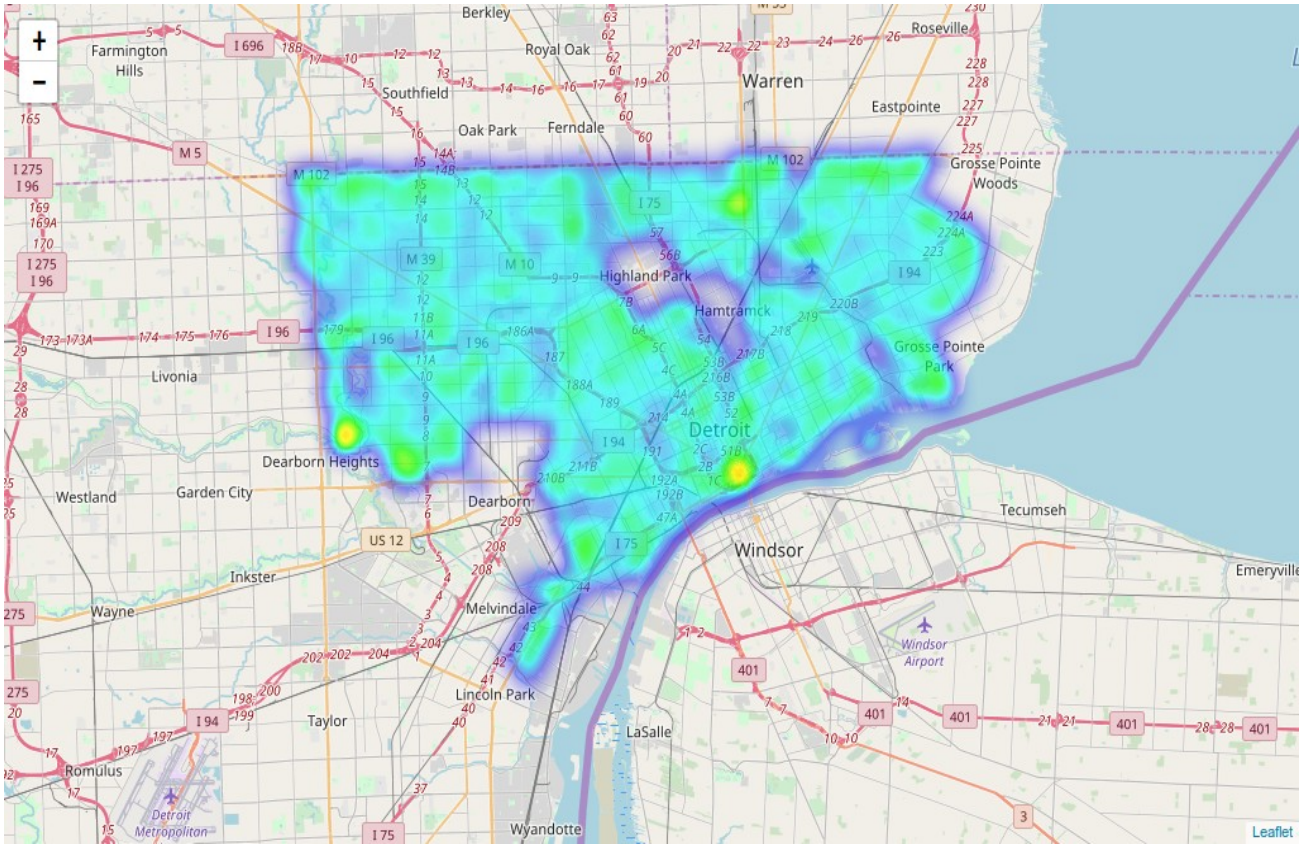
```
crime_data['building_id'] = [pgh.encode(crime_data['LAT'][i],crime_data['LON'][i]) for i in crime_data.index]
```

2.2. Visualization

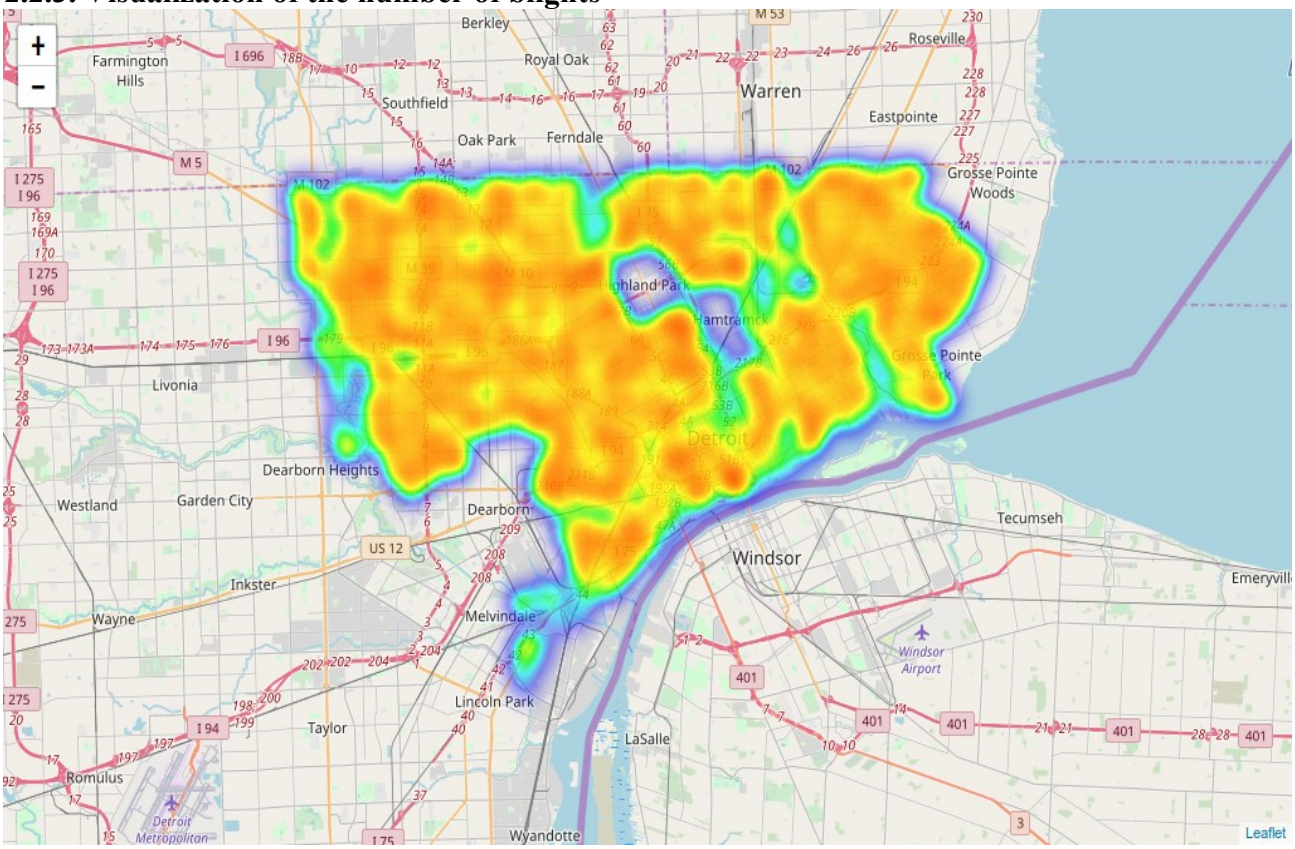
2.2.1. Visualization of the number of crimes



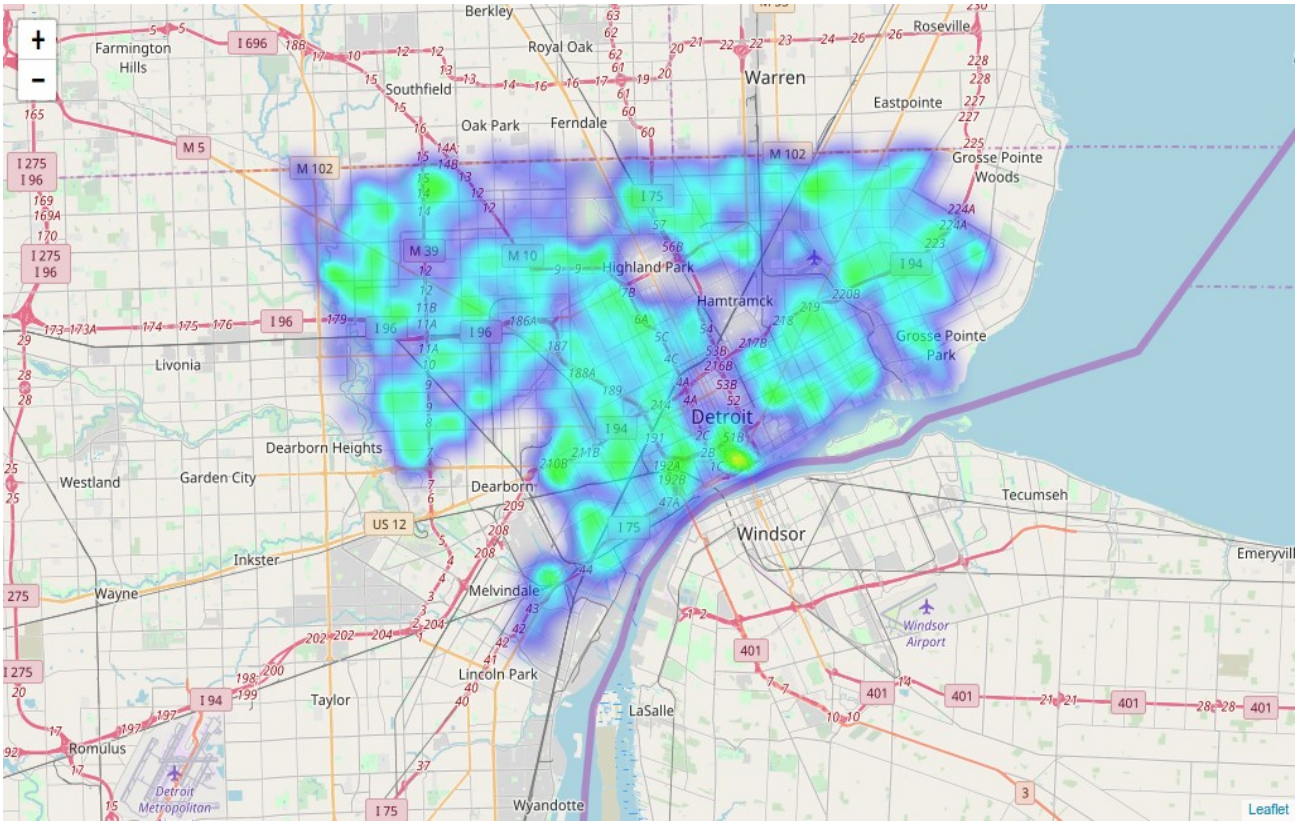
2.2.2. Visualization of the number of 311 calls



2.2.3. Visualization of the number of blights



2.2.4. Visualization of the number of demolition permits



2.2.5. Observation

At a glance, demolition permit data is correlated with 311 calls data, on the other hand, crime data and blight data are not.

2.3. Feature Engineering

One dimensional input, two classes classification

Here, I show features and classes for my one dimensional input two classes classification models. There are three models:

1. feature: the number of blights
   classes: blighted, non-blighted

2. feature: the number of 311calls
   classes: blighted, non-blighted

3. feature: the number of crimes
   classes: blighted, non-blighted

Multi dimensional input, two classes classification

Here, I show features and classes for my multi dimensional input two classes classification model. There is one model:

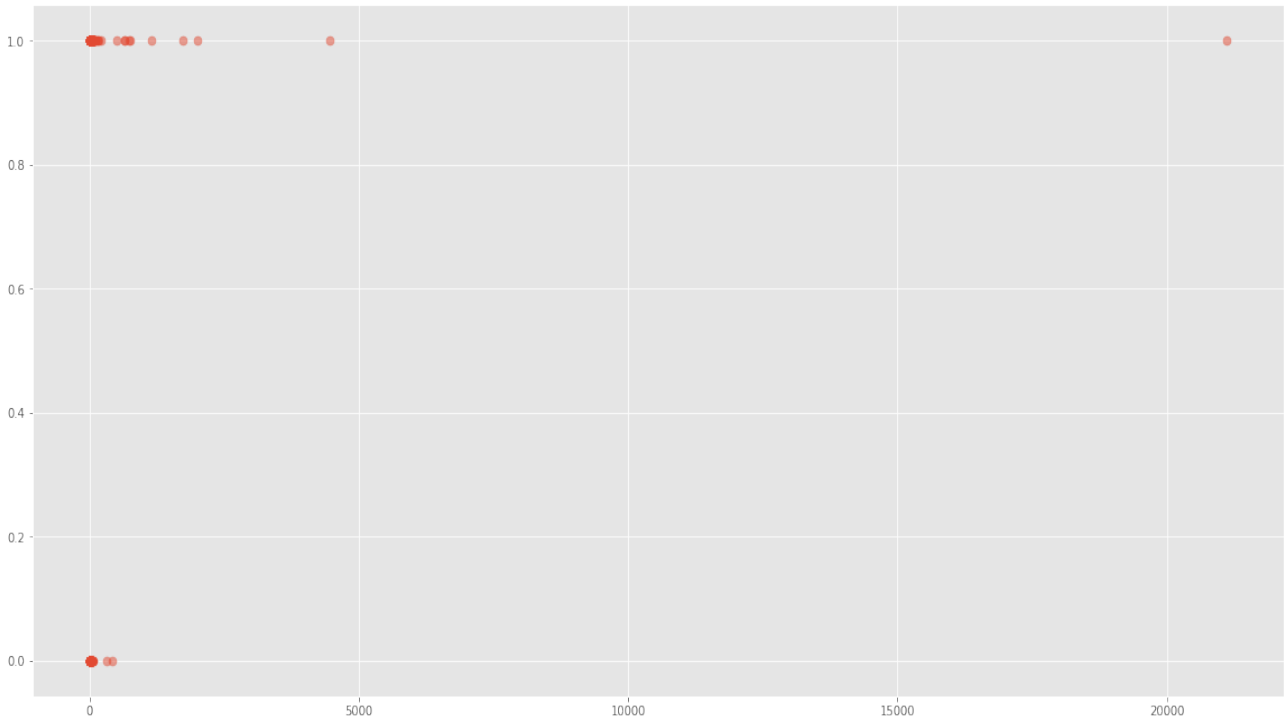
features: the number of blights, number of 311calls, number of crimes
classes: blighted, non-blighted

2.4. Classification and Prediction

One dimensional input, two classes classification

1. feature: number of blight violations

horizontal axis: the number of blight violations
vertical axis: whether blighted (= 1) or not blighted (= 0)

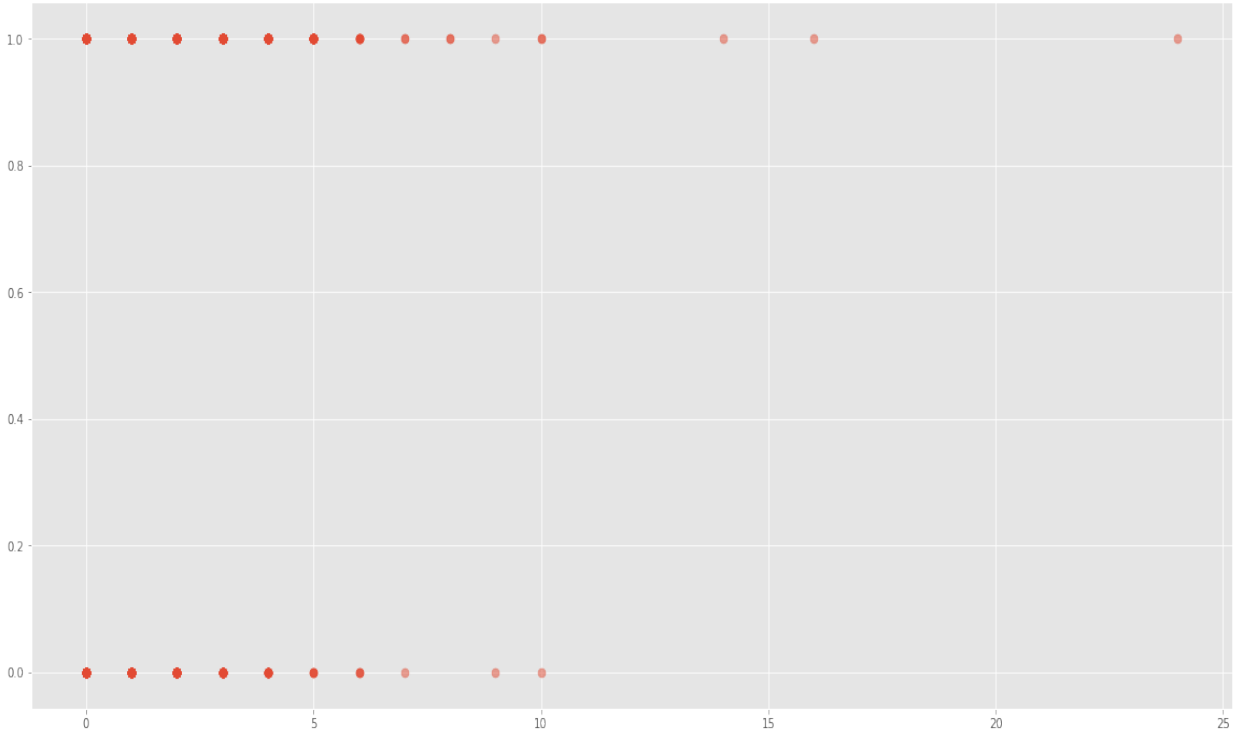


The result of analysis

Method	**Accuracy**
Logistic Regression	0.8101
Boosting	0.8784
Random Forest	0.8784

2. feature: number of 311calls

horizontal axis: the number of 311 calls
vertical axis: whether blighted (= 1) or not blighted (= 0)

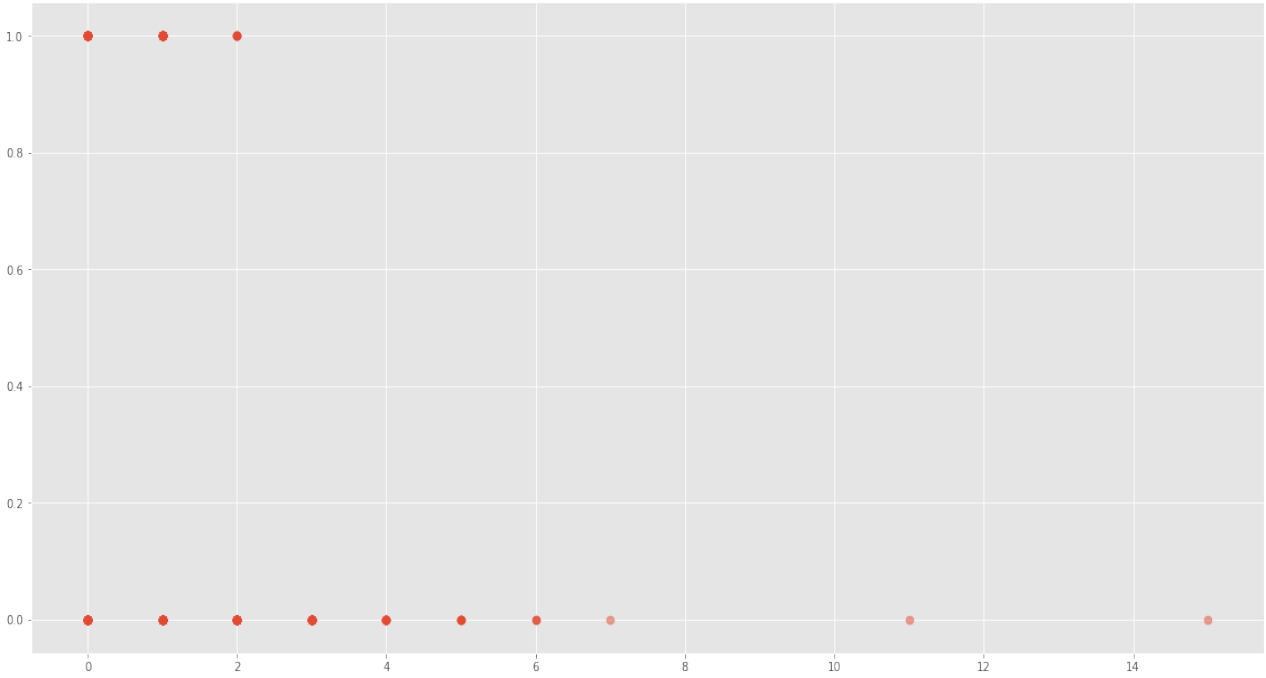


The result of analysis

Method	**Accuracy**
Logistic Regression	0.6235
Boosting	0.6235
Random Forest	0.6235

3. feature: number of crimes

horizontal axis: the number of crimes in a region
vertical axis: whether blighted (= 1) or not blighted (= 0)



The result of analysis

Method	**Accuracy**
Logistic Regression	0.8928
Boosting	0.9093
Random Forest	0.9093

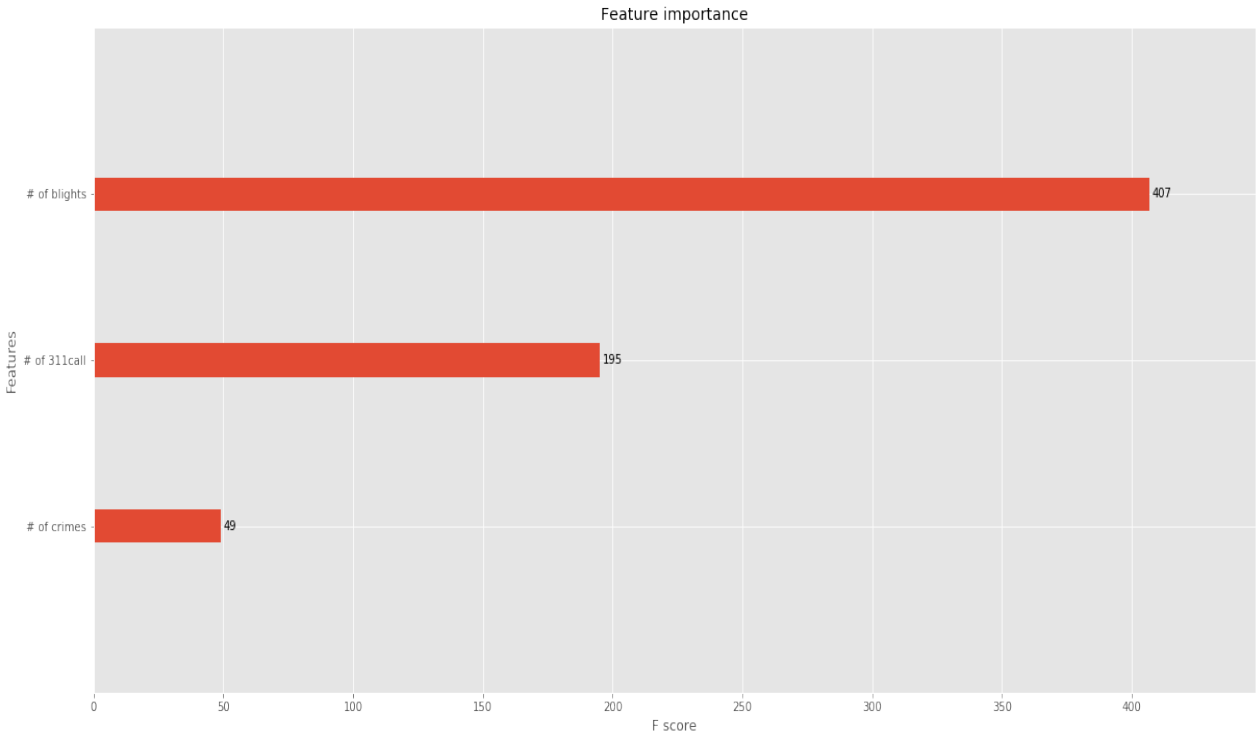
Multi dimensional input, two classes classification

1. features: number of blights, number of 311calls, number of crimes

The result of analysis

Method	**Accuracy**
Logistic Regression	0.8933
Boosting	0.9146
Random Forest	0.9093

Feature importances in boosting



3. Discussion and Conclusion

3.1. Issues and future work in this project

It would be possible to improve accuracy of prediction by appropriately choosing features. The most significant issue is that I did not include the structure of time and space in our model. Morckel reports housing abandonment is highly correlated with time and space. In this project, I did not employ features on time and space; however, the given data include such features. Including those features into our model would improve accuracy and prediction ability of the model.

In addition, adding more features could improve our model. It is not sure why but Kaggle winners report a lot of features for test data improve prediction rate. At least, adding features on time and space must be significant.

Moreover, other approaches could be used for improvement such as Deep Learning. For this project, employing Deep Learning using some libraries such as Keras would be valid. It is very easy to use such a library, however some critical issues exist. For example, to analyze big data, high-spec computer or preparing for Cloud environment such as AWS, GCP, Azure should be needed.

Furthermore, using a framework Dash must be helpful when deploying dashboard. Dash helps build reactive application, and the application helps managers make decision with interactive explorer of data.

Finally, how to remove outliers should be improved. Because boundaries of Detroit were decided roughly, some outliers are not removed. Those points should be removed.

3.2. Conclusion

I could finally find a good model using three features in Boosting. The model is already a respectable predictor, therefore we might be able to use this predictor to improve Detroit; however, we have to keep on improving it using data collected in the future because the city is changing. Data driven project is just getting started. Data analysts have to keep on working and improving their work from now on.

Reference

Morckel, V. C. (2014). Spatial characteristics of housing abandonment. *Applied Geography*, *48*, 8-16.