Machine Learning Engineer Nanodegree

Capstone Project

Alex Dementsov
July 15, 2020

Project Overview

Traffic accidents involving vehicle crashes is an inevitable part of developed society. They cause people injuries and deaths, damage to property and infrastructure. There are many factors that contribute to the crashes, including human factors, environmental conditions and infrastructure.

Analysis of vehicle crashes is valuable for insurance companies to estimate risks and come up with competitive insurance premiums, identify possible fraud and intentional crash events.

Problem Statement

In this project a street clustering based on vehicle crashes in New York City is investigated. Number of injured and killed people during vehicle crashes as well as contributing factors and vehicle types define streets. Because there are no labels involved, this is an unsupervised learning problem. The goal is to cluster streets by similarity in terms of crashes and identify features that contribute most. I use aggregated data from original features over a certain period of time (2012-2019) and derivative features to cluster the streets.

Metrics

Mean Squared Error (MSE) is used for evaluation metrics.

Data Exploration

In the project I use the public NYC dataset: "Motor Vehicle Collisions - Crashes" for 2012-2019 which has about **1.64M** of rows.
https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95

The original features used in training are the following:

Feature name	**Description**
CRASH DATE	Occurrence date of crash

CRASH TIME	Occurrence time of crash
LATITUDE	Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)
LONGITUDE	Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)
NUMBER OF PERSONS INJURED	Number of persons injured
NUMBER OF PERSONS KILLED	Number of persons killed
NUMBER OF PEDESTRIANS INJURED	Number of pedestrians injured
NUMBER OF PEDESTRIANS KILLED	Number of pedestrians killed
NUMBER OF CYCLIST INJURED	Number of cyclists injured
NUMBER OF CYCLIST KILLED	Number of cyclists killed
NUMBER OF MOTORIST INJURED	Number of vehicle occupants injured
NUMBER OF MOTORIST KILLED	Number of vehicle occupants killed
CONTRIBUTING FACTOR VEHICLE 1	Factors contributing to the collision for designated vehicle
VEHICLE TYPE CODE 1	Type of vehicle based on the selected vehicle category (ATV, bicycle, car/suv, ebike, escooter, truck/bus, motorcycle, other)
PERSON_SEX	(from Person file) Gender of victim
VEHICLE_MAKE	(from Vehicle file) Vehicle make

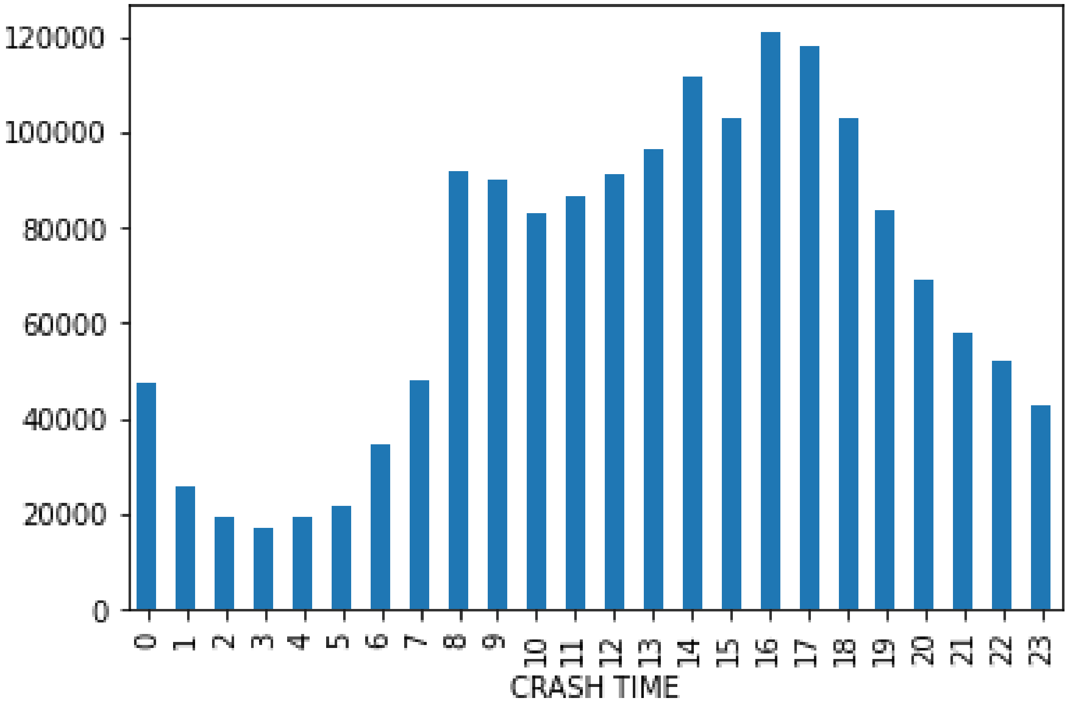
First 5 rows:

```
  CRASH DATE CRASH TIME   LATITUDE  LONGITUDE  NUMBER OF PERSONS KILLED  \
0  06/20/2018      17:25  40.805820 -73.909260                       0.0
1  07/09/2018       9:20  40.842750 -73.924774                       0.0
```

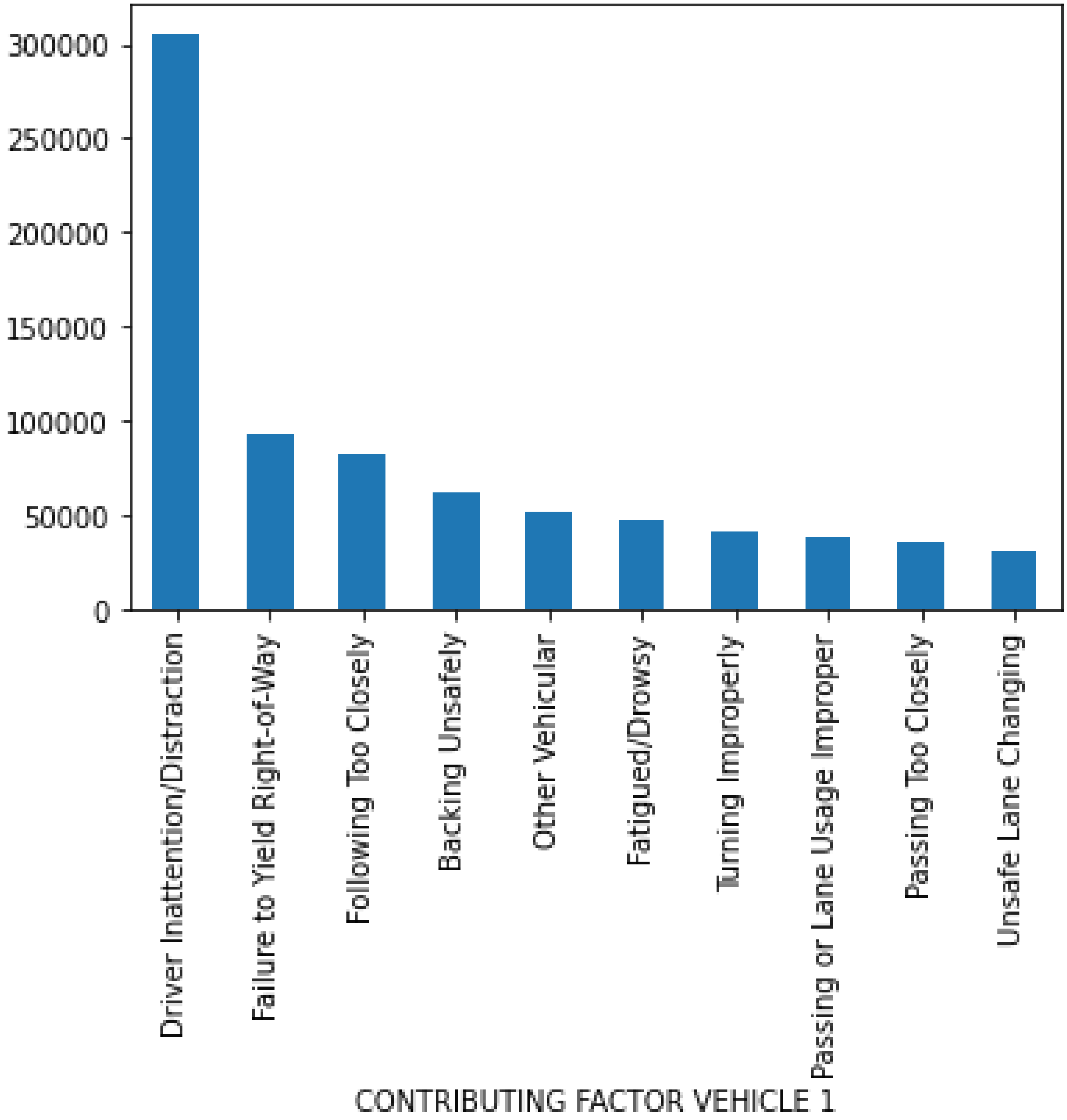
```
2  06/28/2018      12:06  40.847180 -73.921350                       0.0
3  07/11/2018      22:15  40.698864 -73.909890                       0.0
4  06/20/2018      11:50  40.748302 -73.978350                       0.0

   NUMBER OF PEDESTRIANS INJURED  NUMBER OF PEDESTRIANS KILLED  \
0                              0                             0
1                              0                             0
2                              0                             0
3                              0                             0
4                              0                             0

   NUMBER OF CYCLIST INJURED  NUMBER OF CYCLIST KILLED  \
0                          0                         0
1                          0                         0
2                          0                         0
3                          0                         0
4                          0                         0

   NUMBER OF MOTORIST INJURED  NUMBER OF MOTORIST KILLED  \
0                           0                          0
1                           0                          0
2                           0                          0
3                           0                          0
4                           0                          0

    CONTRIBUTING FACTOR VEHICLE 1                  VEHICLE TYPE CODE 1
0  Driver Inattention/Distraction  Station Wagon/Sport Utility Vehicle
1  Passing or Lane Usage Improper                                Sedan
2                     Unspecified                                Sedan
3  Passing or Lane Usage Improper  Station Wagon/Sport Utility Vehicle
4  Driver Inattention/Distraction                                Sedan
```

Here are some charts that give some sense of data.

Number of crashes by hour.


120000
100000
80000
60000
40000
20000
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
CRASH TIME

Top 10 contributing factors:



Algorithms and Techniques

In the project I cluster streets data by using PCA and K-means algorithms.
I aggregate data over the 2012-2019 period of time and perform geo join with New York OpenStreetMap.

The feature set used in calculations:

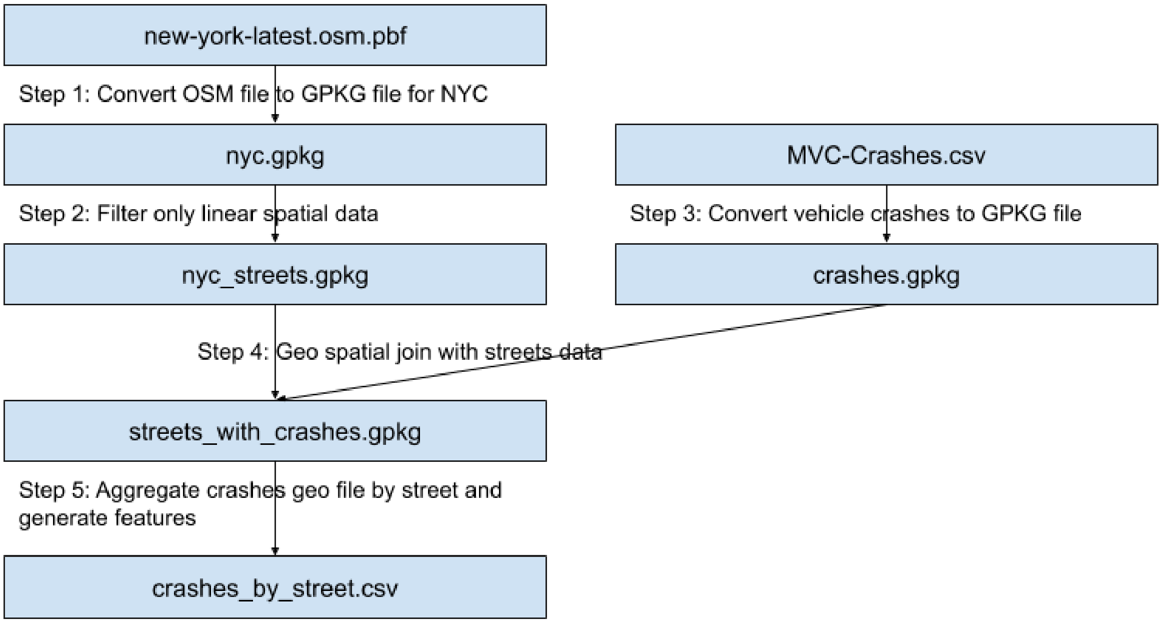
- Street name
- Number of with injury, killed for persons, pedestrians, cyclist, motorist (2*4=8 features)
- Contributing factors (top 10 features)
- Vehicle types (top 10 features)
- Weekday (7 features)
- Is weekend
- Hour (24 features)
- Month (12 features)

Benchmark

I compare results with New York City street highway types.

Data Processing

The workflow for the preprocessing is presented in the diagram:


new-york-latest.osm.pbf
Step 1: Convert OSM file to GPKG file for NYC
nyc.gpkg
MVC-Crashes.csv
Step 2: Filter only linear spatial data
Step 3: Convert vehicle crashes to GPKG file
nyc_streets.gpkg
crashes.gpkg
Step 4: Geo spatial join with streets data
streets_with_crashes.gpkg
Step 5: Aggregate crashes geo file by street and generate features
crashes_by_street.csv

`MVC-Crashes.csv` has street data but it's very incomplete. But it has latitude and longitude that I used to do geo spatial join and associate vehicle crashes with the NYC streets.

Step 1: Convert OSM file to GPKG file for NYC

Input:	new-york-latest.osm.pbf (231 Mb)
Output:	nyc.gpkg (581 Mb)

I used New York OpenStreetMap and converted to Geo package file with `ogr2ogr` tool within New York City bounding box. `nyc.gpkg` file spatial data (points, lines, polygons) as well as metadata associated with the spatial data. We are only interested in linear data where `highway` column is set.

```
$ ogr2ogr -f GPKG nyc.gpkg new-york-latest.osm.pbf -spat -74.309432
40.518589 -73.667336 40.988810
```

Step 2: Filter only linear spatial data

Input:	nyc.gpkg (581 Mb)
Output:	nyc_streets.gpkg (29 Mb)

I manually extracted linear spatial data using the QGIS tool by filtering features with highway column set and stored in `nyc_streets_raw.gpkg`. Then I used script to generate `nyc_streets.gpkg` file. The script only uses data with values `highway = ['trunk', 'primary', 'secondary', 'tertiary', 'residential', 'motorway', 'unclassified']`. Because street name may be not unique (e.g. Main Street) I need to associate it with the city. I used `tiger:county` column to get the city name.

```
$ python3 created_streets.py
```

Step 3: Convert vehicle crashes to GPKG file

Intput:	MVC-Crashes.csv (354 Mb)
Output:	crashes.gpkg (256 Mb)
	contributing_factors.csv (1.6 Kb)
	vehicle_types.csv (6 Kb)

Before I associate crash location with a street I need to convert vehicle crashes data to Geo package file.

```
$ python3 create_crashes_geofile.py
```

Step 4: Geo spatial join with streets data

Intput:	nyc_streets.gpkg (29 Mb)
	crashes.gpkg (256 Mb)
Output:	streets_with_crashes.gpkg (5.4 Gb) - gigantic file (!)

At this step we associate crash location with a street name by performing geo spatial join. Before geo join operation is performed we add buffer of 10 meters to the streets geo data.

```
$ python3 create_geojoin_file.py
```

Step 5: Aggregate crashes geo file by street and generate features

Input:	streets_with_crashes.gpkg (5.4 Gb)
Output:	crashes_by_street.csv (24 Kb)

At this step I use geo joined file `streets_with_crashes.gpkg` and generate a file with 78 original and engineered features that is used for clustering algorithm.

```
$ python3 generate_data.py
```

The script does the following:

- Adds 10 top contributing factors
- Adds 10 top vehicle types factors
- Adds datetime features: hours, months, weekdays and weekends
- Adds is intersection or is not intersection crashes features

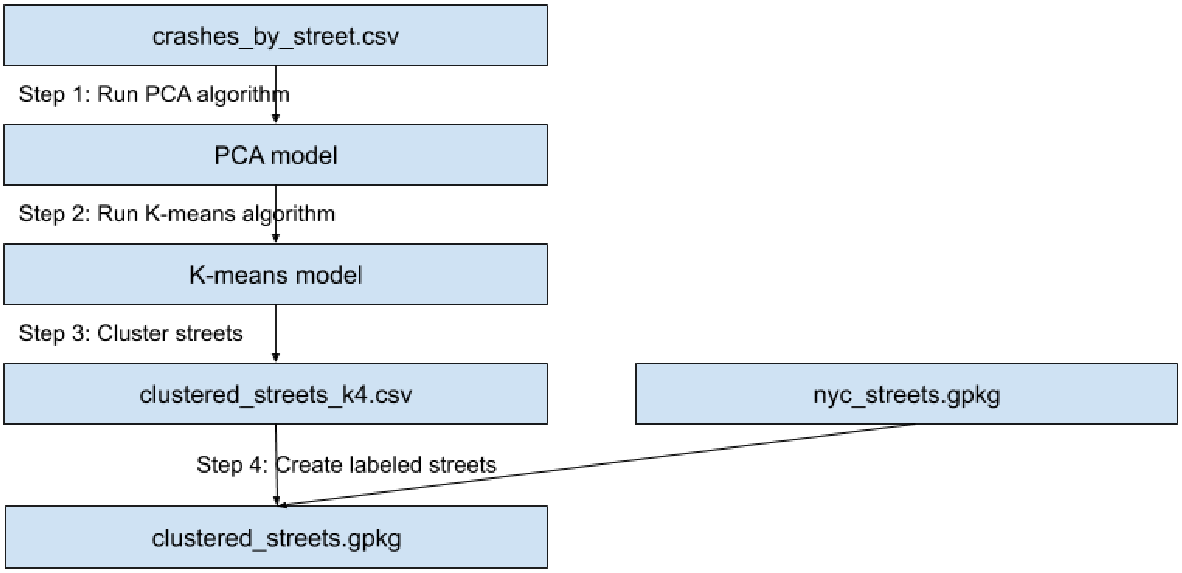
First few lines of the file are displayed below:

```
$ head -n 5 crashes_by_street.csv

street_city,number_of_persons_injured,number_of_persons_killed,number
_of_pedestrians_injured,number_of_pedestrians_killed,number_of_cyclis
t_injured,number_of_cyclist_killed,number_of_motorist_injured,number_
of_motorist_killed,factor__driver_inattention_distraction,factor__fai
lure_to_yield_right_of_way,factor__following_too_closely,factor__back
ing_unsafely,factor__other_vehicular,factor__fatigued_drowsy,factor__
turning_improperly,factor__passing_or_lane_usage_improper,factor__pas
sing_too_closely,factor__unsafe_lane_changing,factor__traffic_control
_disregarded,factor__driver_inexperience,vehicle__passenger_vehicle,v
ehicle__suv,vehicle__sedan,vehicle__taxi,vehicle__van,vehicle__pick_u
p_truck,vehicle__bus,vehicle__bicycle,vehicle__ambulance,vehicle__tra
ctor,vehicle__motorcycle,hour_0,hour_1,hour_2,hour_3,hour_4,hour_5,ho
ur_6,hour_7,hour_8,hour_9,hour_10,hour_11,hour_12,hour_13,hour_14,hou
r_15,hour_16,hour_17,hour_18,hour_19,hour_20,hour_21,hour_22,hour_23,
monday,tuesday,wednesday,thursday,friday,saturday,sunday,is_weekend,j
anuary,february,march,april,may,june,july,august,september,october,no
vember,december,is_intersection,is_not_intersection
"100th Avenue, Queens,
NY",97,2,3,0,4,0,90,2,41,24,3,7,7,1,1,7,1,0,7,4,125,46,37,0,1,4,2,1,0
,0,0,5,4,1,6,1,3,7,13,27,14,7,10,10,8,21,7,14,17,11,7,9,6,6,6,25,37,3
5,38,27,34,24,58,19,12,15,15,18,16,26,21,16,17,21,24,202,18
```

```
"100th Drive, Queens,
NY",4,0,0,0,1,0,3,0,3,1,0,1,1,0,0,0,1,0,0,1,7,6,2,0,0,0,0,1,0,0,0,1,0
,0,0,0,0,0,0,2,0,2,0,1,2,2,1,0,2,0,0,1,0,0,3,3,3,3,2,3,3,0,3,4,1,2,1,
0,0,1,1,3,0,0,4,13,4
"100th Road, Queens,
NY",2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
1,0,0,0,0,0,0,0,1,0
"100th Street, Kings,
NY",8,0,2,0,1,0,5,0,7,3,2,4,1,1,5,3,1,0,1,0,45,20,5,0,1,2,1,0,0,0,0,5
,0,2,1,0,0,2,0,3,5,5,5,3,7,4,6,7,5,4,2,5,4,3,4,14,13,9,12,15,10,9,19,
7,6,6,11,10,3,10,6,6,8,3,6,0,82
"100th Street, Queens,
NY",189,0,48,0,10,0,131,0,182,35,7,22,14,4,7,2,5,3,16,9,283,198,95,10
,10,17,2,1,3,1,4,16,4,8,6,19,11,14,26,55,35,39,36,28,49,45,50,49,50,4
4,26,20,23,12,10,103,101,88,99,108,92,84,176,68,59,66,39,50,49,59,60,
60,63,55,47,546,129
```

Implementation

Most implementation details are described in the "NYC Vehicle Crashes" jupyter notebook. Here is the data flow of the implementation.

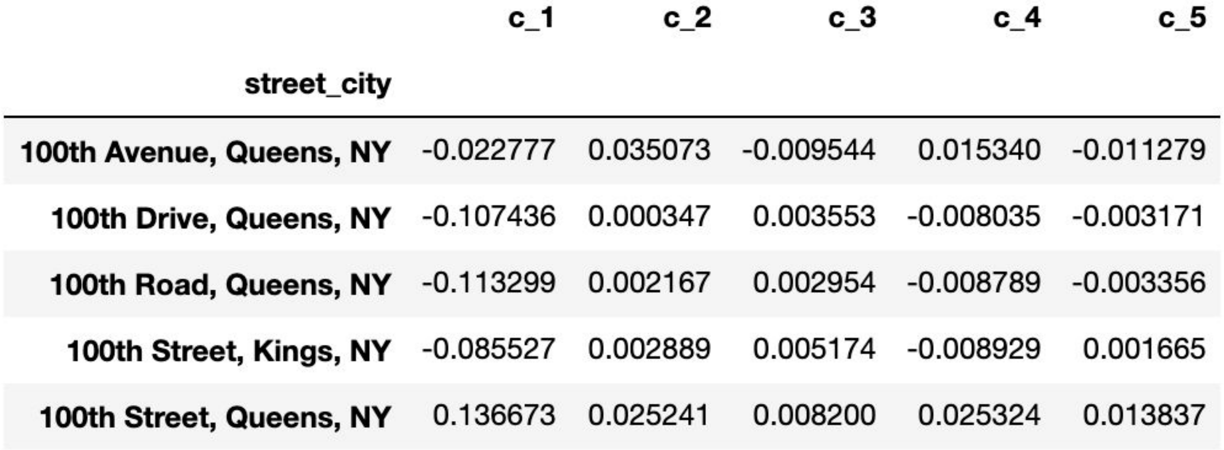


Step 1: Run PCA algorithm

Input: crashes_by_street.csv

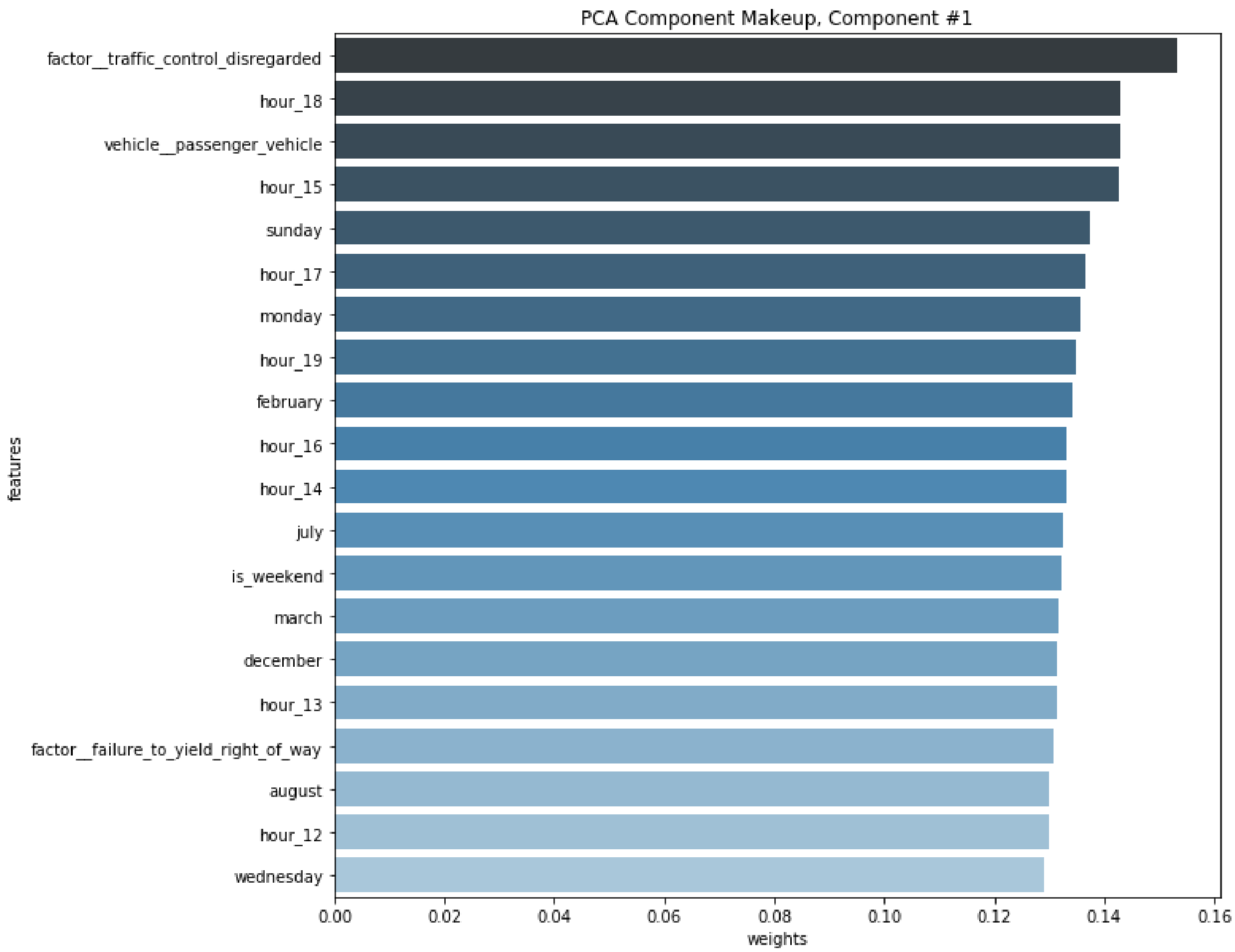
Output: PCA model.tar.gz (22.3 Kb)
transformed streets data (in memory)

At this step I use `crashes_by_street.csv` to run a PCA algorithm (sagemaker.PCA) to find principal components. This step is needed to reduce dimensionality of the problem. Five (5) PCA components will give 95.5% data variance. Here is the first 5 rows represented in 5 components:



	c_1	c_2	c_3	c_4	c_5
street_city					
100th Avenue, Queens, NY	-0.022777	0.035073	-0.009544	0.015340	-0.011279
100th Drive, Queens, NY	-0.107436	0.000347	0.003553	-0.008035	-0.003171
100th Road, Queens, NY	-0.113299	0.002167	0.002954	-0.008789	-0.003356
100th Street, Kings, NY	-0.085527	0.002889	0.005174	-0.008929	0.001665
100th Street, Queens, NY	0.136673	0.025241	0.008200	0.025324	0.013837

Component #1 represented as a superposition of features.

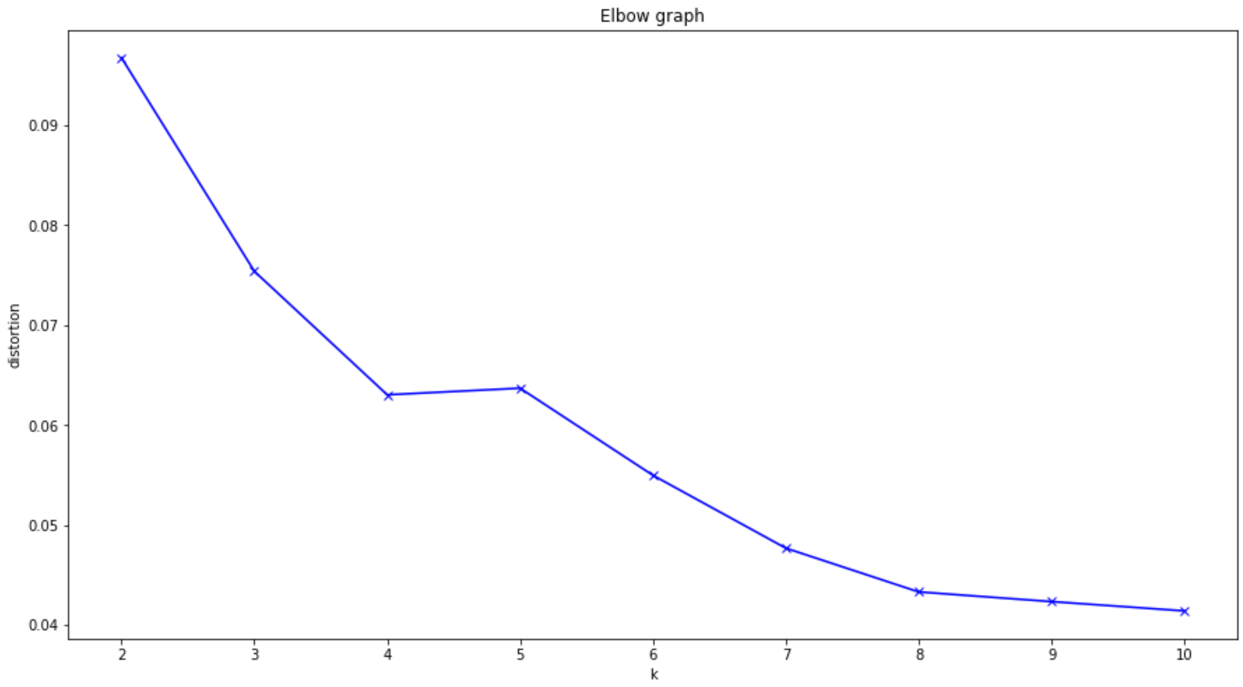


Component #1is the most important component as it gives 88.1% data variance (for simplicity we could just use this one component). As you can see the most important feature in component #1 is "Traffic control disregarded". The other contributing factor is "Failure to yield right of way". There is one vehicle feature: "Passenger vehicle". Other features are mostly related to hour, weekday and month: hours 18, 15 and 17 (6pm, 3pm and 5pm), Sunday and Monday. My guess is that this is the time and day that cause most vehicle crashes. People are driving tired after work or getting drunk.

Step 2: Run K-means algorithm

Input: transformed streets data (in memory)
Output: K-means model.tar.gz (1.4 Kb)

With the PCA model I transformed the streets data and used it for K-means model. To find an optimal K I ran K-means algorithm for k in range 2 to 10 and calculated disturbance (sum of distances to cluster centers) and created an elbow graph which shows that K=4 is an optimal K.



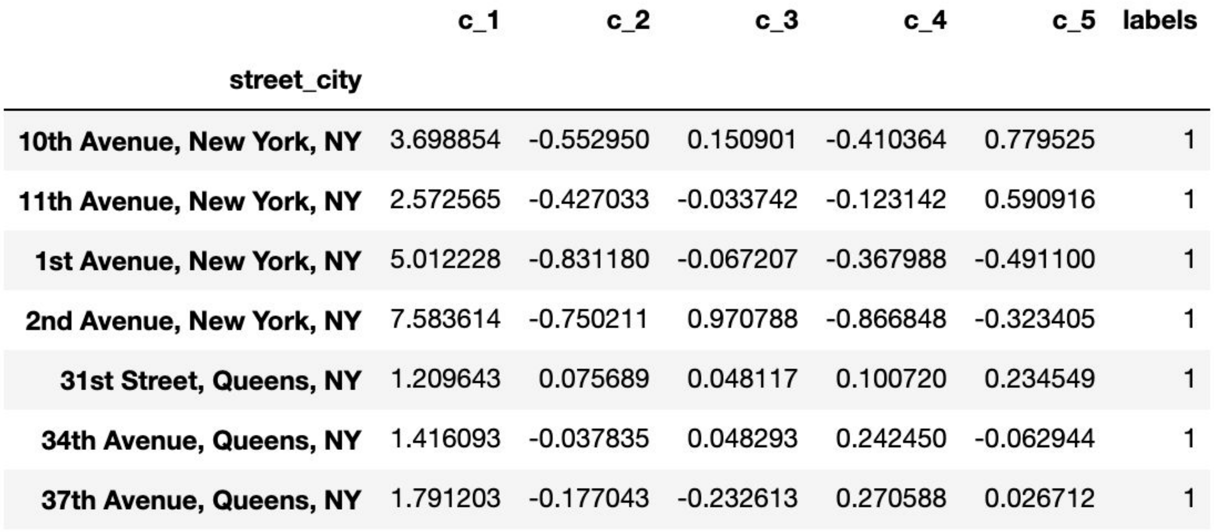
Step 3: Cluster streets

Input: K-means model.tar.gz (1.4 Kb)
Output: clustered_streets_k4.csv (1.1 Mb)

User K-means model to label cluster streets. Here is the number of streets in each cluster:

Cluster label	Number of streets
1	6407
3	1141
0	383
2	59

First 5 rows of the output file `clustered_streets_k4.csv`:



	c_1	c_2	c_3	c_4	c_5	labels
street_city						
10th Avenue, New York, NY	3.698854	-0.552950	0.150901	-0.410364	0.779525	1
11th Avenue, New York, NY	2.572565	-0.427033	-0.033742	-0.123142	0.590916	1
1st Avenue, New York, NY	5.012228	-0.831180	-0.067207	-0.367988	-0.491100	1
2nd Avenue, New York, NY	7.583614	-0.750211	0.970788	-0.866848	-0.323405	1
31st Street, Queens, NY	1.209643	0.075689	0.048117	0.100720	0.234549	1
34th Avenue, Queens, NY	1.416093	-0.037835	0.048293	0.242450	-0.062944	1
37th Avenue, Queens, NY	1.791203	-0.177043	-0.232613	0.270588	0.026712	1

Step 4: Create labeled streets

Input: clustered_streets_k4.csv (1.1 Mb)
nyc_streets.gpkg (29 Mb)
Output: clustered_streets.gpkg (12 Mb)

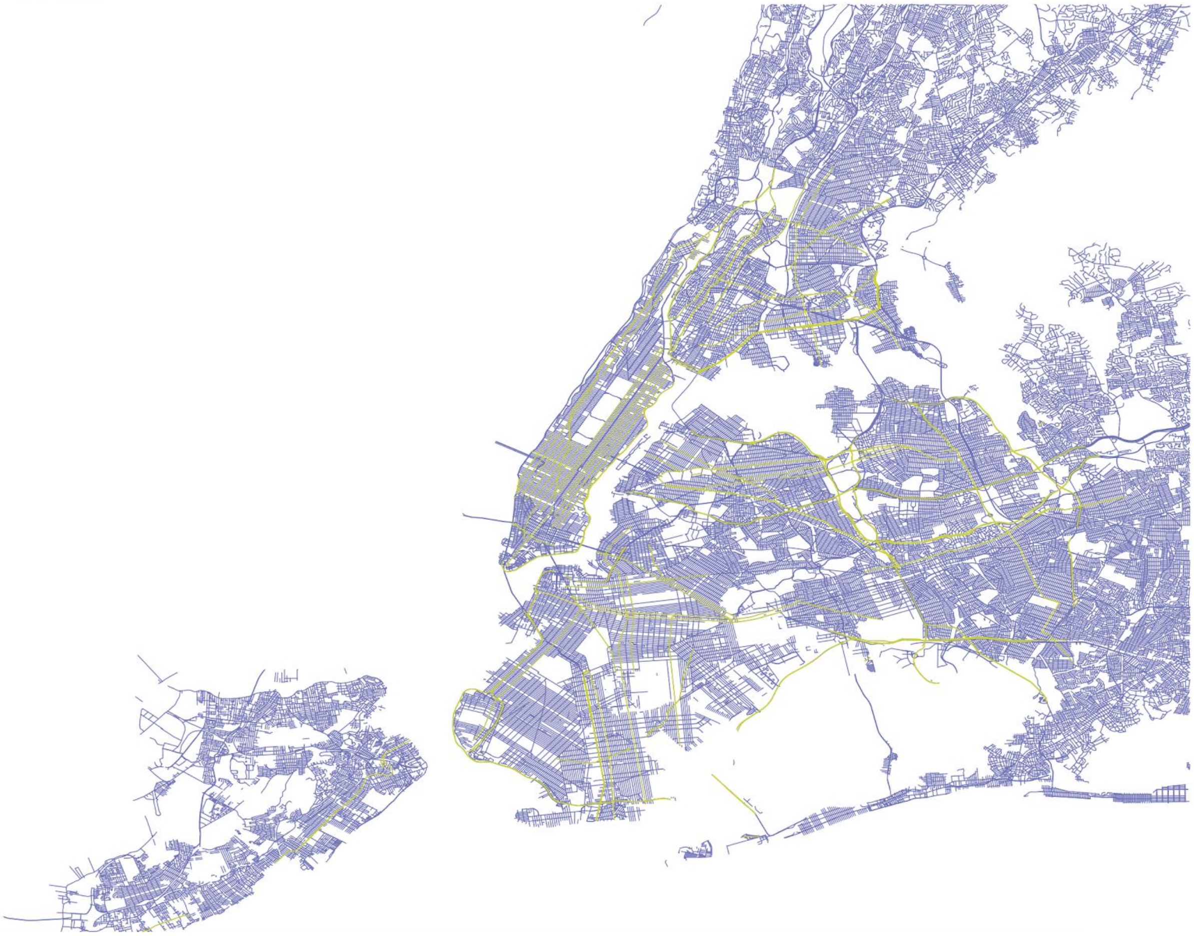
Final step is to create a `clustered_streets.gpkg` which merges labeled data with streets geo spatial data.

Results

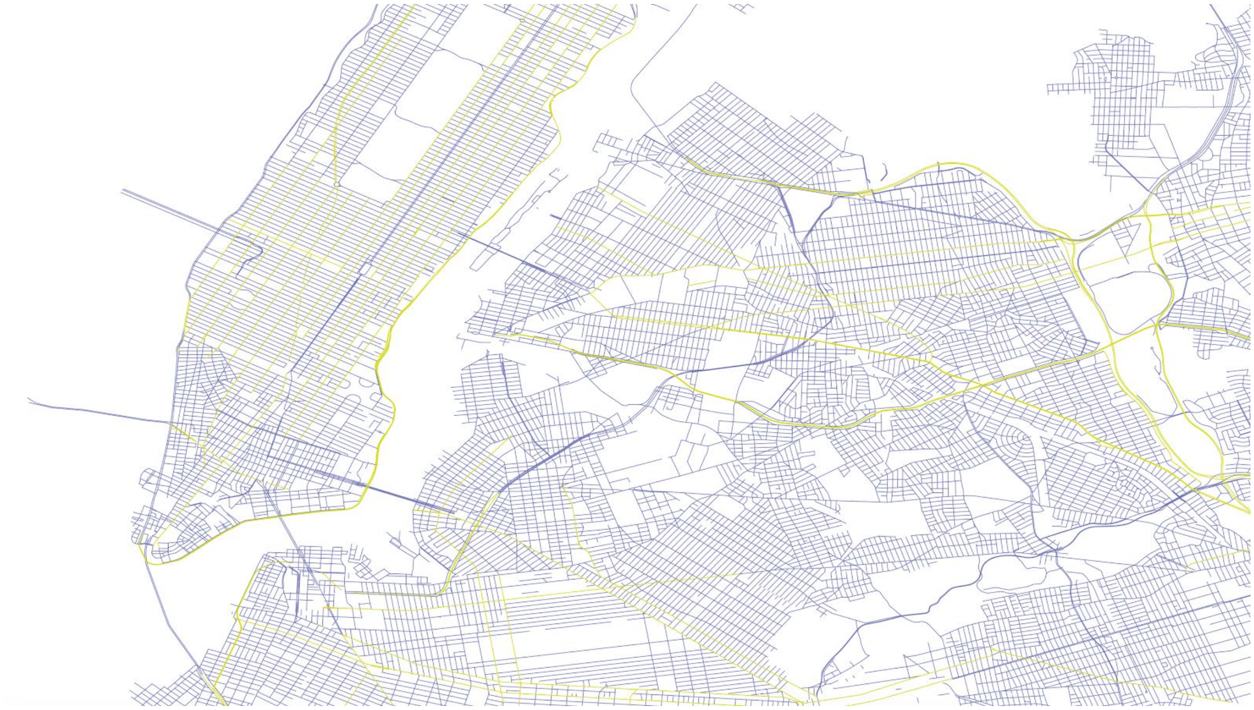
The clustered streets can be displayed on the map. Here I show the streets with cluster #2 and #3.

Cluster #0: High risk streets around Lincoln Tunnel and Brooklyn area around Atlantic Ave.
Cluster #1: Suburban area and Staten Island.
Cluster #2: Large highways, avenues and waterfronts.
Cluster #3: Streets on Manhattan (perpendicular to avenues) and Brooklyn.

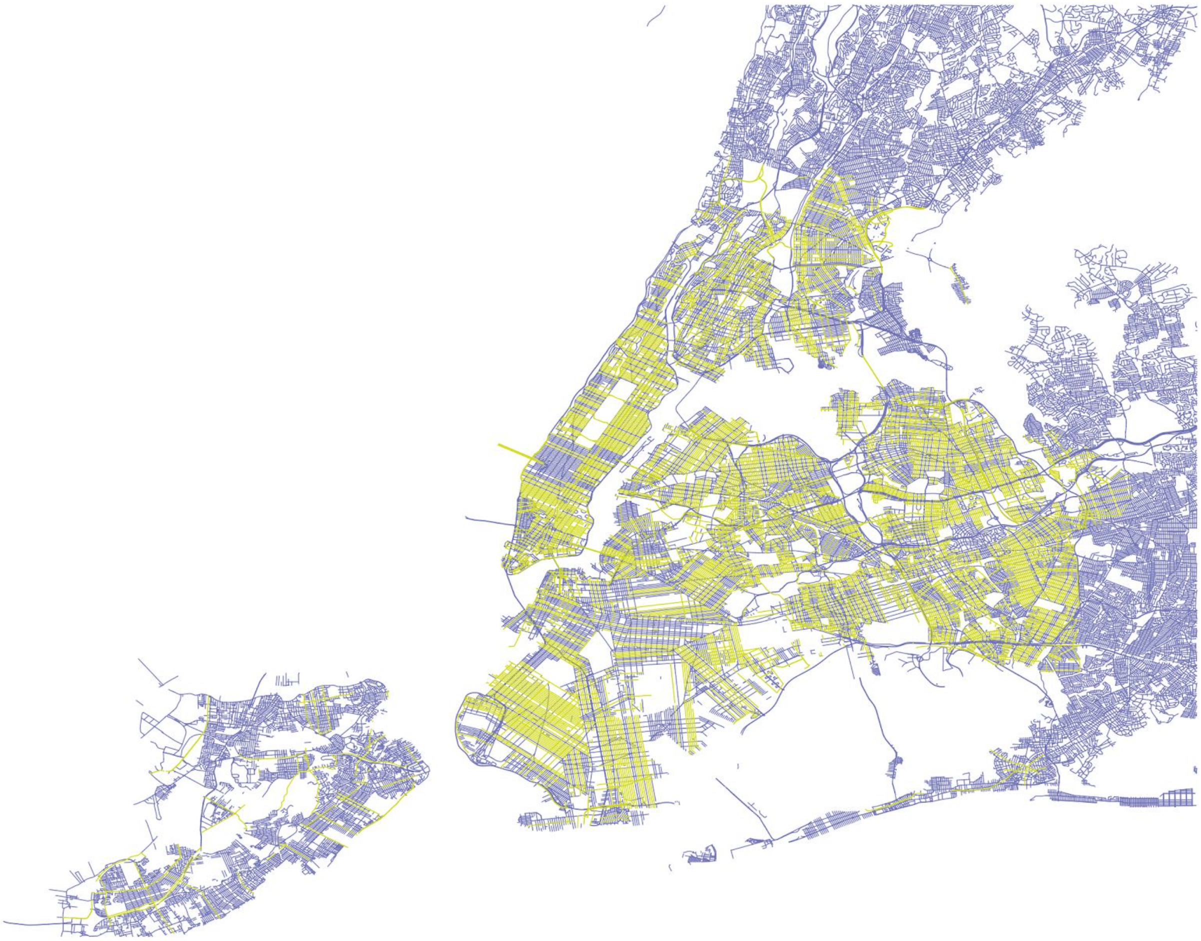
Cluster #2



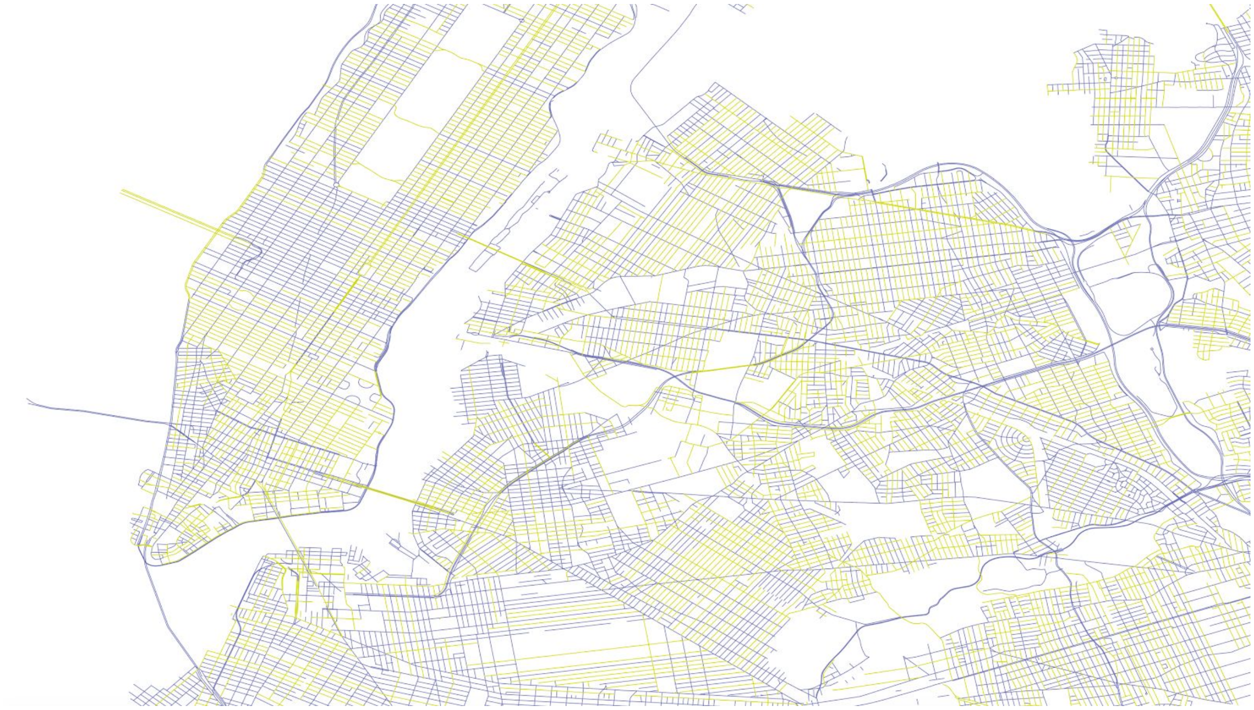
Cluster #2 zoomed in



Cluster #3



Cluster #3 zoomed in



Future Considerations

To improve model of vehicle crashes additional factors can be taken into consideration:

1. City places (e.g. restaurants, stores)
One assumption is that proximity to liquor stores or stadiums can increase crashes. The places can be geospatially joined with crashes locations. Possible datasets include "Google Places API" (https://developers.google.com/places/web-service/search) or similar.

2. NYC events.
One assumption is that large gatherings of people cause traffic jams and eventually crashes.

3. Weather precipitation.
One assumption is that high precipitation (rain or snow) can increase crashes.

Additional interesting analysis can include vehicles and people involved in the crashes and parking violations.

"Motor Vehicle Collisions - Vehicles"
https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Vehicles/bm4k-52h4
"Motor Vehicle Collisions - Person"

https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Person/f55k-p6yu
"Parking Violations Issued - Fiscal Year 2020"
https://data.cityofnewyork.us/City-Government/Parking-Violations-Issued-Fiscal-Year-2020/pvqr-7yc4

Note: Due to Covid-19 Q4 2019 and onward data show anomaly (5x decrease in vehicle crashes) and are out of scope of thi s analysis.