Coursera Capstone Project

Week 5 – Final Project

<u>Crime prediction in Chicago</u>

Final course of IBM Data Science Professional Certificate by Coursera

1 - Introduction

Many world metropolises have overpopulation and high purchasing power in common. This mixture of ingredients often leads to high crime rates. Since tourists do not have an in-depth knowledge of common crime points, this project was inspired by the protection of these tourists.

Real world case

In general, every traveler, whether for business or pleasure, intends to enjoy a little of the place and have fun. Inspired by a scenario in which some co-workers will go to Chicago on business, they will have a few more days to explore the city. Knowing them, I know they like to go out at night after work and drink some drikes and listen to good music.

This project's main idea is to predict the potential for a crime to happen close to a nightclub search on foursquare.

Let's go!

2 - Data

In this session I will quickly explain the origin and method of data acquisition.

Crimes occurred in Chicago in 2019

The data were accessed at: https://data.cityofchicago.org/Public-Safety/Crimes-2019/w98m-zvie, where the API was generated to download in https://data.cityofchicago.org/api/views/w98m-zvie/rows.csv?accessType=DOWNLOAD&api_foundry=true.

Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Yea
JC222395	04/12/2019 07:15:00 AM	085XX S INGLESIDE AVE	0266	CRIMINAL SEXUAL ASSAULT	PREDATORY	SCHOOL - PUBLIC BUILDING	false	false	0632	006	8	44	02	1184042	1848646	201
JC198380	03/23/2019 08:30:00 AM	093XX S SANGAMON ST	1752	OFFENSE INVOLVING CHILDREN	AGGRAVATED CRIMINAL SEXUAL ABUSE BY FAMILY MEMBER	RESIDENCE	false	true	2223	022	21	73	17	1171628	1842756	201
JC176179	03/06/2019 12:40:00 PM	064XX S WHIPPLE ST	0266	CRIMINAL SEXUAL ASSAULT	PREDATORY	RESIDENCE	true	true	0823	008	17	66	02	1157161	1861685	201
JC128860	01/25/2019 03:00:00 AM	101XX S LAFAYETTE AVE	0266	CRIMINAL SEXUAL ASSAULT	PREDATORY	RESIDENCE	true	true	0511	005	9	49	02	1177711	1837787	201
JC358783	07/21/2019 07:20:00 PM	022XX N LONG AVE	041A	BATTERY	AGGRAVATED - HANDGUN	RESIDENCE - PORCH / HALLWAY	false	false	2515	025	36	19	04B	1140015	1914457	201

After some procedures for cleaning the data and creating new columns (which can be seen in detail in the attached notebook, remember, we are in the "Readme"), a new dataframe was created for the crimes that occurred in 2019.

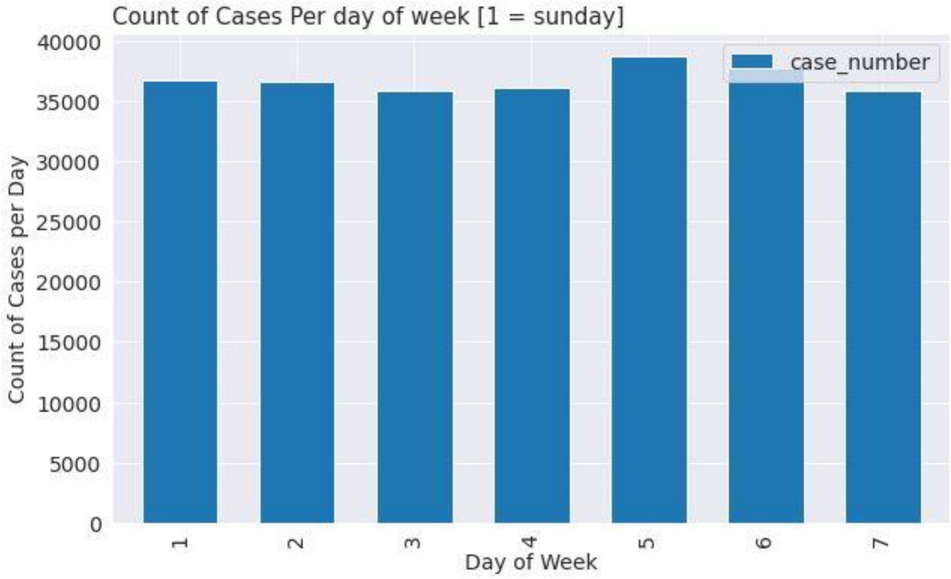
```
[ ] crime_2019['date'] = pd.to_datetime(crime_2019['date'], format='%m/%d/%Y %I:%M:%S %p')
```

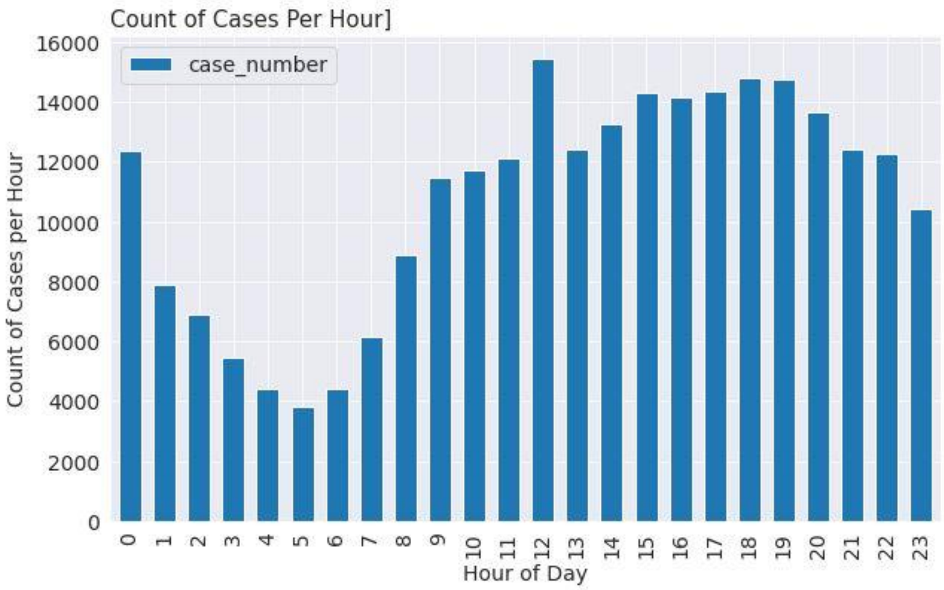
```
[ ] # Add new columns to the dataframe to allow hourly, daily & monthly analysis
    crime_2019['hour'] = crime_2019['date'].dt.hour
    crime_2019['day_name'] = crime_2019['date'].dt.day_name()
    crime_2019['day'] = crime_2019['date'].dt.dayofweek + 1
    crime_2019['month_name'] = crime_2019['date'].dt.month_name()
    crime_2019['month'] = crime_2019['date'].dt.month
    crime_2019['year'] = crime_2019['date'].dt.year
    crime_2019['year_month'] = crime_2019['date'].dt.to_period('M')

    # Add the zip and street attributes
    crime_2019['zip'] = crime_2019.block.str.split(' ').str[0]
    crime_2019['street'] = crime_2019.block.str.split(' ').str[1:].apply(', '.join)

    crime_2019.head()
```

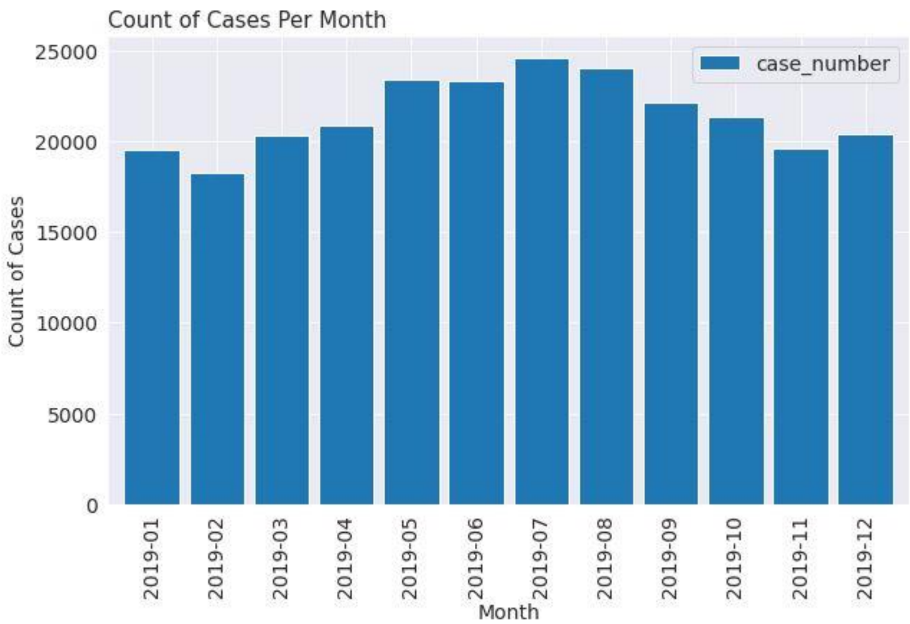
	case_number	date	block	primary_type	ward	latitude	longitude	hour	day_name	day	month_name	month	year	year_month	zip	street
0	JC222395	2019-04-12 07:15:00	085XX S INGLESIDE AVE	CRIMINAL SEXUAL ASSAULT	8.0	41.739860	-87.601274	7	Friday	5	April	4	2019	2019-04	085XX	S, INGLESIDE, AVE
1	JC198380	2019-03-23 08:30:00	093XX S SANGAMON ST	OFFENSE INVOLVING CHILDREN	21.0	41.723978	-87.646929	8	Saturday	6	March	3	2019	2019-03	093XX	S, SANGAMON, ST
2	JC176179	2019-03-06 12:40:00	064XX S WHIPPLE ST	CRIMINAL SEXUAL ASSAULT	17.0	41.776227	-87.699411	12	Wednesday	3	March	3	2019	2019-03	064XX	S, WHIPPLE, ST
3	JC128860	2019-01-25 03:00:00	101XX S LAFAYETTE AVE	CRIMINAL SEXUAL ASSAULT	9.0	41.710207	-87.624797	3	Friday	5	January	1	2019	2019-01	101XX	S, LAFAYETTE, AVE
4	JC358783	2019-07-21 19:20:00	022XX N LONG AVE	BATTERY	36.0	41.921370	-87.760978	19	Sunday	7	July	7	2019	2019-07	022XX	N, LONG, AVE

Descriptive statistics of Chicago crimes

Graphs

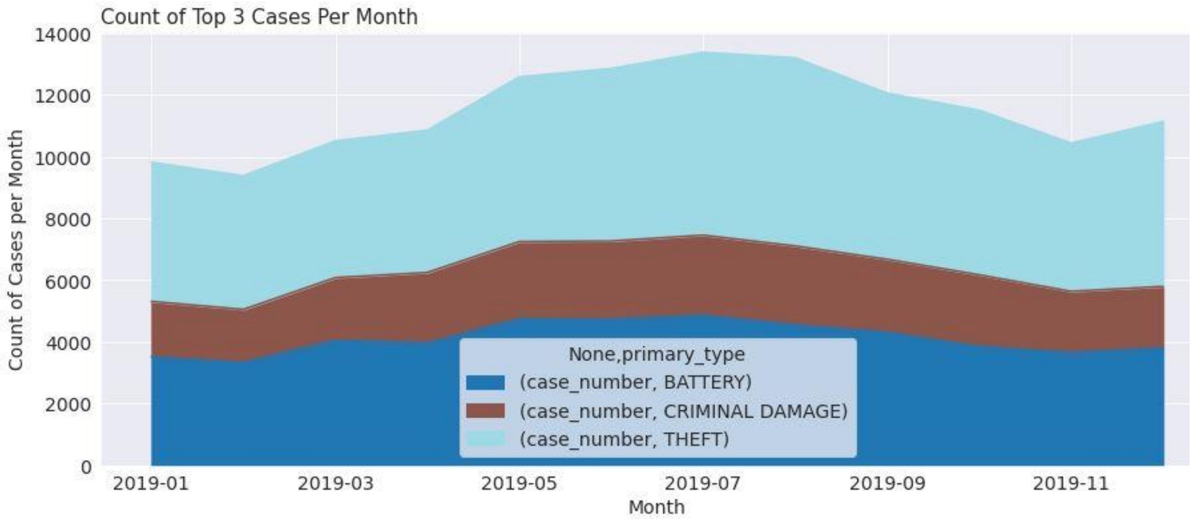


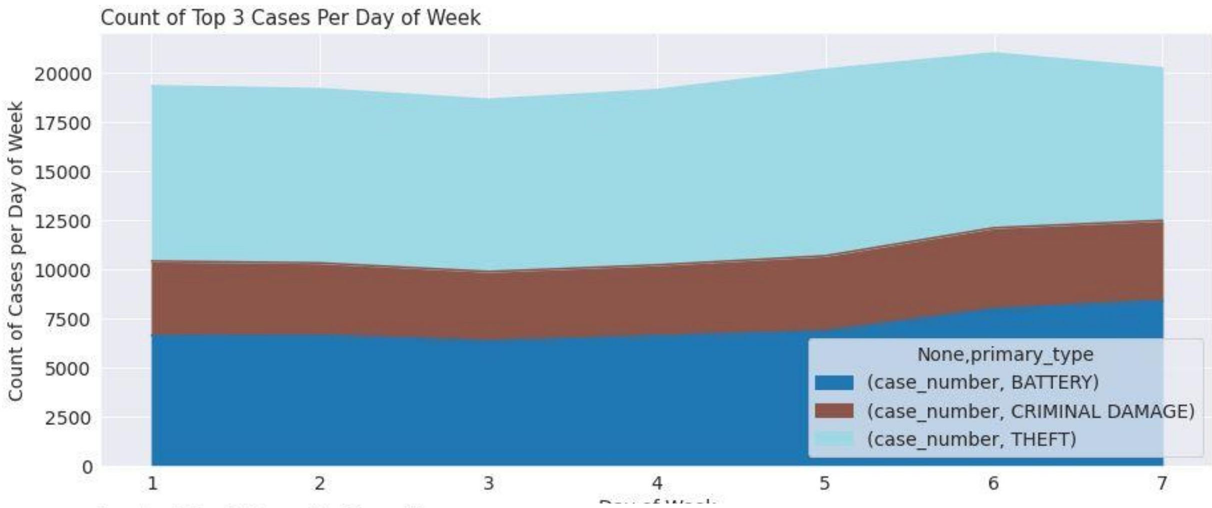


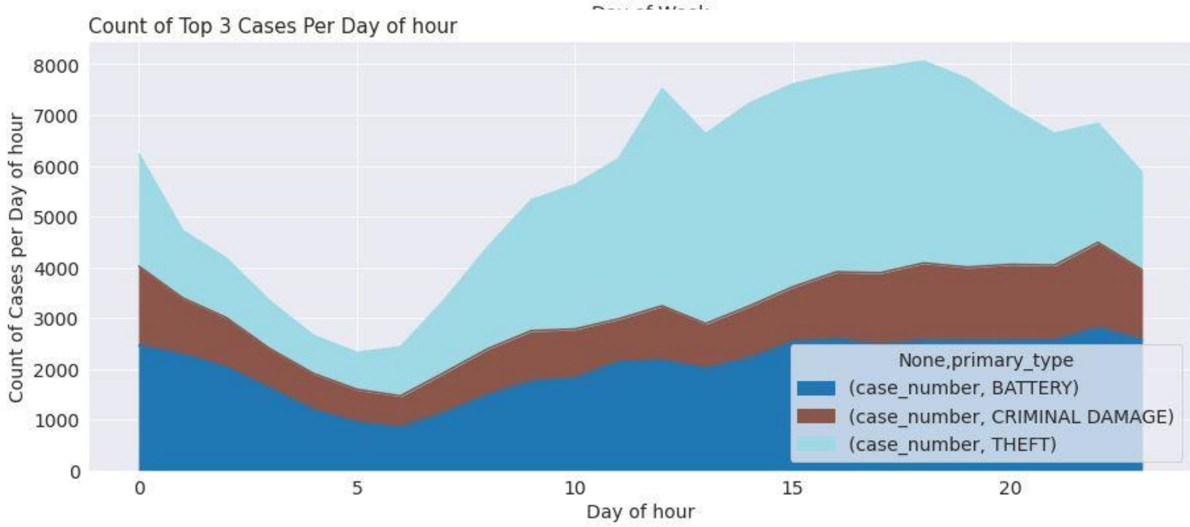


```python
# Crimes are the 3 most commonly occuring ones
crime_2019[['primary_type', 'case_number']].groupby(
    ['primary_type'], as_index=False).count().sort_values(
    'case_number', ascending=False).head(3)
```

	primary_type	case_number
30	THEFT	61612
2	BATTERY	49460
6	CRIMINAL DAMAGE	26597

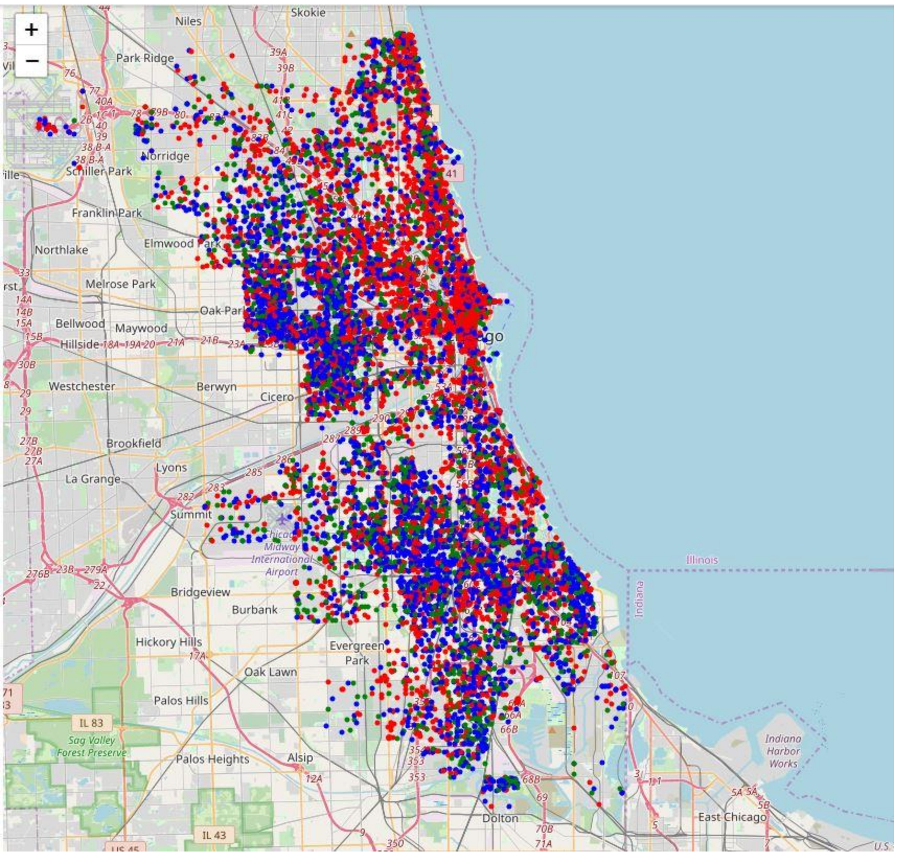


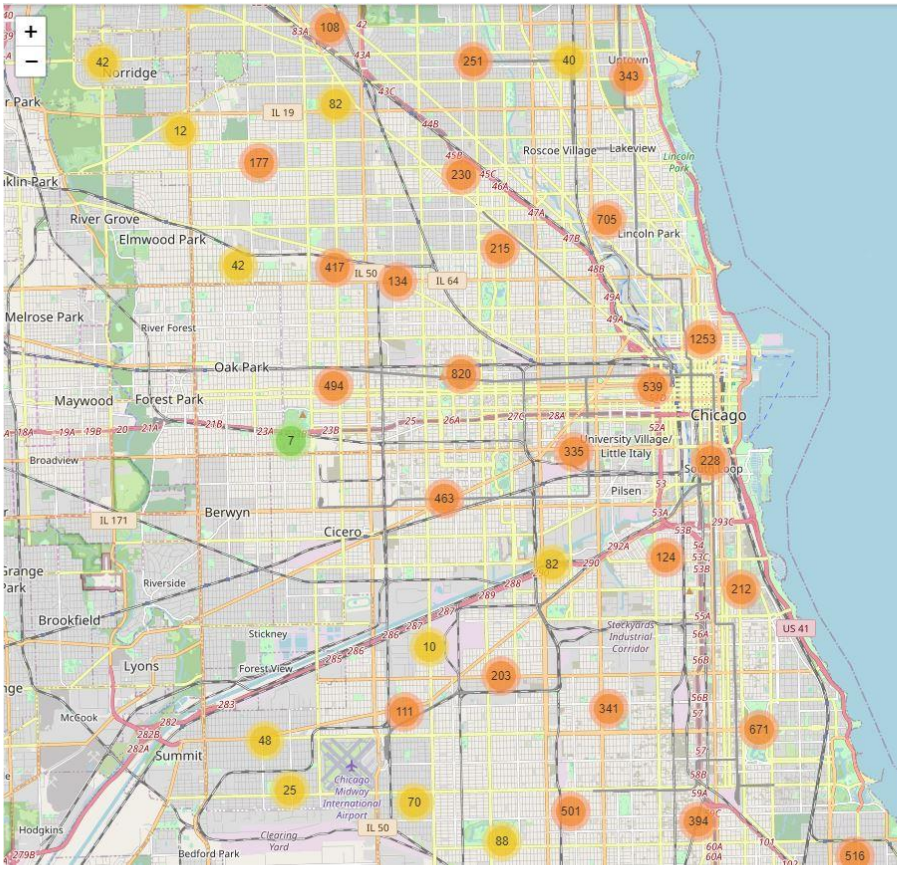


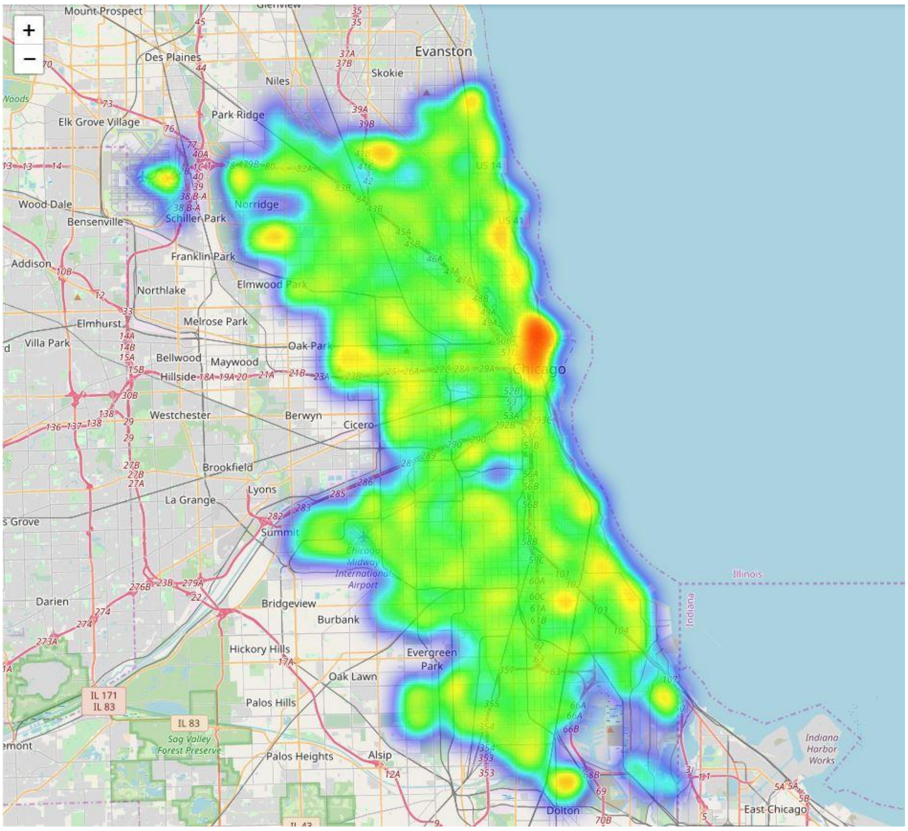


Maps

Distribution of the 3 most common types of crimes in Chicago in july or reasons of ease of processing data.







Nightclubs

With a focus on a particular region where there are many blogs talking about nightlife, the data was taken by the link below, where it was focused on a particular region and nightlife. https://foursquare.com/explore?mode=url&ne=41.893525%2C-87.622678&q=Nightlife&sw=41.886624%2C-87.635788. The data was retrieved by a foursquare API where my credentials were introduced and the link above, where it was converted to HTML where I took the most important "classes". You can see more details in the attached notebook.

id	score	name	address	postalcode	city	href	latitude	longitude	category	likes
554e1fb8498eafc2606cb999	8.8	Broken Shaker	19 E Ohio St	60611	Chicago	/v/broken-shaker/554e1fb8498eafc2606cb999	41.892488	-87.627336	Cocktail	300
5774640fcd100a944f17f4e9	8.5	London House Rooftop Bar	85 E Wacker Dr	60601	Chicago	/v/london-house-rooftop-bar/5774640fcd100a944f...	41.888031	-87.625264	Hotel Bar	111
4b7ecfe4f964a520470130e3	9.2	Gilt Bar	230 W Kinzie St	60654	Chicago	/v/gilt-bar/4b7ecfe4f964a520470130e3	41.889236	-87.635377	Bar	464
51ca4b11498eb327cdb072b1	8.8	The Berkshire Room	15 E Ohio St	60611	Chicago	/v/the-berkshire-room/51ca4b11498eb327cdb072b1	41.892495	-87.627476	Bar	239
5736ab1f498e4ad304e06cfa	8.8	Raised Rooftop at Renaissance Hotel	1 W Wacker Dr	60601	Chicago	/v/raised-rooftop-at-renaissance-hotel/5736ab1...	41.886726	-87.628110	Bar	104

Maps

Initially the top 3 nightclubs were selected based on the likes received on foursquare, however, later those 3 were insufficient, with that, I used the top 30 nightclubs.

score	name	address	postalcode	city	href	latitude	longitude	category	likes
8.6	Three Dots and a Dash	435 N Clark St	60654	Chicago	/v/three-dots-and-a-dash/51f7183b8bbdc6a6ae21592e	41.890270	-87.630690	Tiki Bar	917
7.7	Public House	400 North State Street	60654	Chicago	/v/public-house/4ca4d2a4d695199c9a9451e7	41.889474	-87.628274	Bar	575
9.2	Gilt Bar	230 W Kinzie St	60654	Chicago	/v/gilt-bar/4b7ecfe4f964a520470130e3	41.889236	-87.635377	Bar	464

Heat map of crimes and nightclubs

3 - Methodology

Modeling

Transform dataframa in only numerical and remove descriptive columns

See more details on the notebook

```
[ ] # Start by copying the Latitude and Longitude to the new DataFrame
    df_features = crime_2019[['latitude', 'longitude']]

    # Next and One Hot Encoding of the hour, day and month variables
    df_features = df_features.join(pd.get_dummies(crime_2019.hour, prefix='hour'))
    df_features = df_features.join(pd.get_dummies(crime_2019.day_name))
    df_features = df_features.join(pd.get_dummies(crime_2019.month_name))

    # Finally add the ward & crimes column, copied from the original Primary Description column
    df_features['ward'] = crime_2019[['ward']]
    df_features['crimes'] = crime_2019[['primary_type']]
```

```
[ ] df_features.head(10)
```

	latitude	longitude	hour_0	hour_1	hour_2	hour_3	hour_4	hour_5	hour_6	hour_7	hour_8	hour_9	hour_10	hour_11	hour_12	hour_13	hour_
0	41.739860	-87.601274	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
1	41.723978	-87.646929	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
2	41.776227	-87.699411	0	0	0	0	0	0	0	0	0	0	0	0	1	0	

Models

Modelling task was turned into a simple binary classification task by only modelling based on the top two most occuring crimes. For each model development 10 Fold Cross Validation was used to ensure the best results were achieved and a Grid Search approach was used to determine the best setting for each of the models

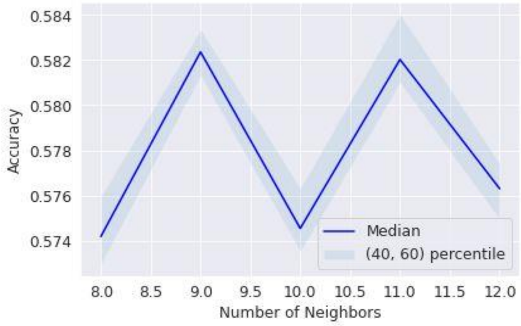
Cross validation

X-Fold Cross Validation

```
[ ] # Function X-Fold Cross Validation
    def cross_validate(model, n_splits = 10):

        k_fold = KFold(n_splits = n_splits)
        scores = [model.fit(X[train], y[train]).score(X[test], y[test]) for train, test in k_fold.split(X)]

        scores = np.percentile(scores, [40, 50, 60])
        return scores
```

K Nearest Neighbours

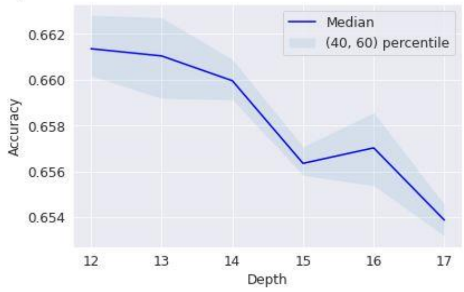
```
Heighbours:  8   2020-04-14 11:39:33.296261
Heighbours:  9   2020-04-14 11:58:57.334631
Heighbours:  10   2020-04-14 12:18:30.476200
Heighbours:  11   2020-04-14 12:37:24.502584
Heighbours:  12   2020-04-14 12:56:40.718368
```




```
[ ] KNN_model = KNeighborsClassifier(n_neighbors = 9).fit(X, y)
```

Decision Three

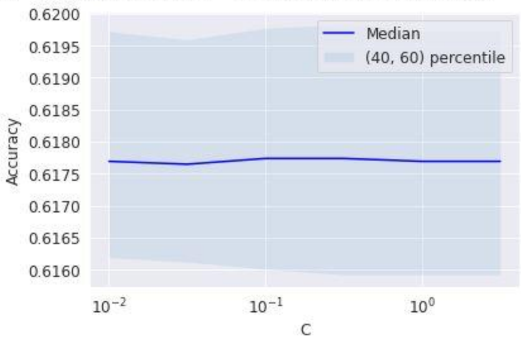
```
Depth:  12   2020-04-14 11:22:37.171828
Depth:  13   2020-04-14 11:22:49.587633
Depth:  14   2020-04-14 11:23:01.754843
Depth:  15   2020-04-14 11:23:13.878943
Depth:  16   2020-04-14 11:23:26.430570
Depth:  17   2020-04-14 11:23:39.539926
```




```
[ ] Tree_model = DecisionTreeClassifier(criterion = "entropy", max_depth = 12).fit(X, y)
```

Logistic Regression

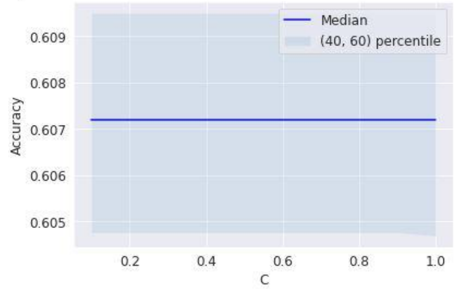
```
C:  0.01   2020-04-14 11:24:01.366071
C:  0.03162277660168379   2020-04-14 11:24:10.169375
C:  0.1   2020-04-14 11:24:18.775032
C:  0.31622776601683794   2020-04-14 11:24:27.451008
C:  1.0   2020-04-14 11:24:36.063854
C:  3.1622776601683795   2020-04-14 11:24:44.727775
```




```
LR_model = LogisticRegression(C = 0.1, solver = 'liblinear').fit(X, y)
```

Naive Bayes

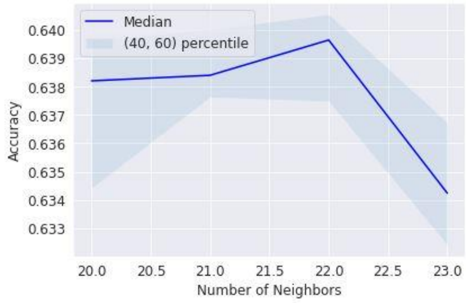
```
Alpha:  0.1    2020-04-14 11:25:58.705666
Alpha:  0.2    2020-04-14 11:26:03.876223
Alpha:  0.30000000000000004   2020-04-14 11:26:09.050006
Alpha:  0.4    2020-04-14 11:26:14.183219
Alpha:  0.5    2020-04-14 11:26:19.371613
Alpha:  0.6    2020-04-14 11:26:24.533896
Alpha:  0.7000000000000001   2020-04-14 11:26:29.713495
Alpha:  0.8    2020-04-14 11:26:34.871969
Alpha:  0.9    2020-04-14 11:26:40.041951
Alpha:  1.0    2020-04-14 11:26:45.203379
```




```
[ ]  NB_model = BernoulliNB(alpha=a).fit(X, y)
```

Decision Forest using a Random Forest

```
Estimator:  20   2020-04-14 11:26:58.111537
Estimator:  21   2020-04-14 11:27:36.607867
Estimator:  22   2020-04-14 11:28:17.126355
Estimator:  23   2020-04-14 11:28:59.199545
```




```
[ ]  Forest_model = RandomForestClassifier(n_estimators = 22, max_features = 'sqrt').fit(X, y)
```

Best model

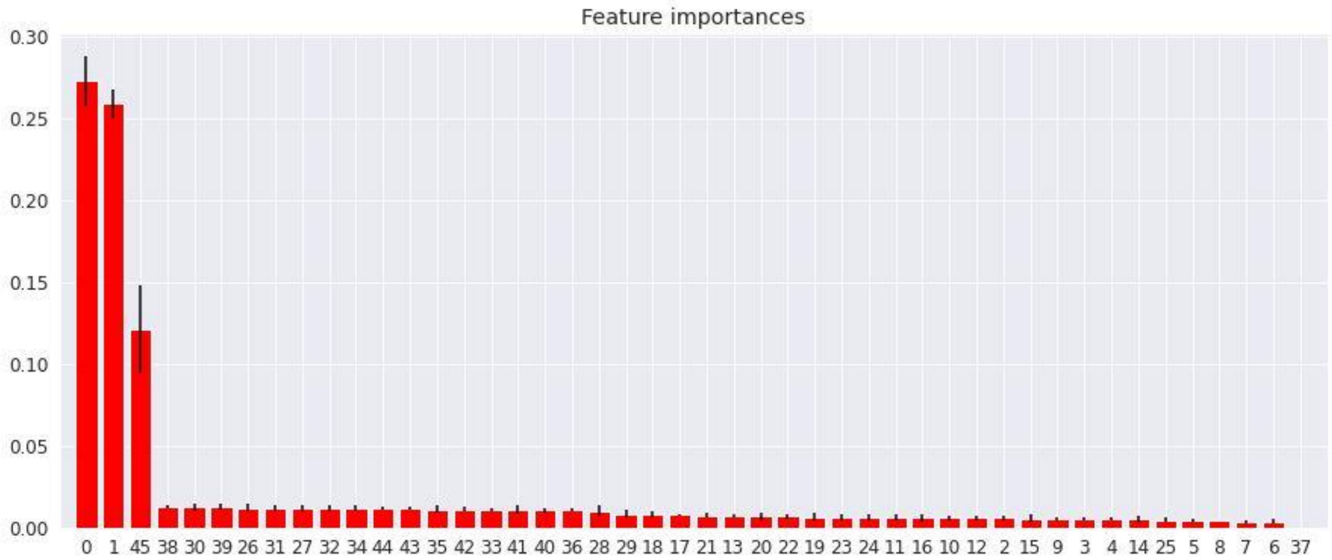
Algorithm	Jaccard	F1-Score	LogLoss
KNN	0.699465	0.729000	10.380224
Decision Tree	0.695396	0.710063	10.520755
Bernoulli Naive Bayes	0.612360	0.664197	13.388800
Logistic Regression	0.621705	0.685819	13.066041
Random Forest	0.995606	0.996037	0.151749

Predict the Final Performance of the Model

```
[ ] # Predict yhat using X_Test
    yhat = Forest_model_final.predict(X_Test)

    # Measure the Jaccard Score of the final Model
    jaccard_final = metrics.jaccard_similarity_score(y_Test, yhat)
    print('Jaccard Score', jaccard_final)

    f1 = metrics.f1_score(y_Test, yhat, average=None)
    print('F1-Score of each class', f1)
```

```
Jaccard Score 0.6454749439042633
F1-Score of each class [0.60123388 0.68087971]
```

Important features



This shows that the most predictive models are:

1. Latitude
2. Longitude
3. Ward

4 - Results

It was not possible to predict where and when crimes will happen, however, a binary combination was made where 0 to represent without crime and 1 to represent potentially that a crime will happen.

▾ Process 1

Fake Crime Data Next we'll generate the fake crime data. The crimes will be equally divided between a no crimes happened 0 and crime happened 1. The Random Forest model will be trainined again on the data from February 2019 to December 2019 and tested against January 2019 to predict the acccuracy of the model.

A new test dataset will then be created for each location in the Top Venues DataFrame and for each Restaurant associated with each of the top Venues. A random visit Date, in January 2019 (view **Create test and train** above), and time will be associated with each row and then a prediction will be made whether a crime would be committed at each location and date or not.

```
[ ] df_features['random_crimes'] = np.random.randint(0, 2, df_features.shape[0])
```

```
[ ] df_features.head()
```

ınday	Thursday	Tuesday	Wednesday	April	August	December	February	January	July	June	March	May	November	October	September	ward	crimes	random_crimes
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	18.0	BATTERY	1
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	35.0	BATTERY	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	6.0	BATTERY	1
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	36.0	BATTERY	1
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	37.0	BATTERY	0

Test Data

The test data was contructed from the the Top Venues Data Frame (nightclubs):

Next a random date and time was assigned to each venue. The date was then split into Hour, Day of Week, Month and Year as described above The data was finally prepared for prediction by applying One Hot encoding and then extracted into a new dataframe that match the format used to create the model. y^ (y_hat) or the predictions were then made

```
yhat
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0])
```

Important - Due to the fact that the top venues (30 night clubs) do not have crimes at all hours or every month, the dataframe created did not have its complete picture, with some columns missing, however, the model was created on top of all data, which have all columns, when generating the forecast, an error occurred in which the model did not have the same number of columns as the top venues data. For this reason, some columns were created manually with values 0 and included manually. See the difference on the notebook.

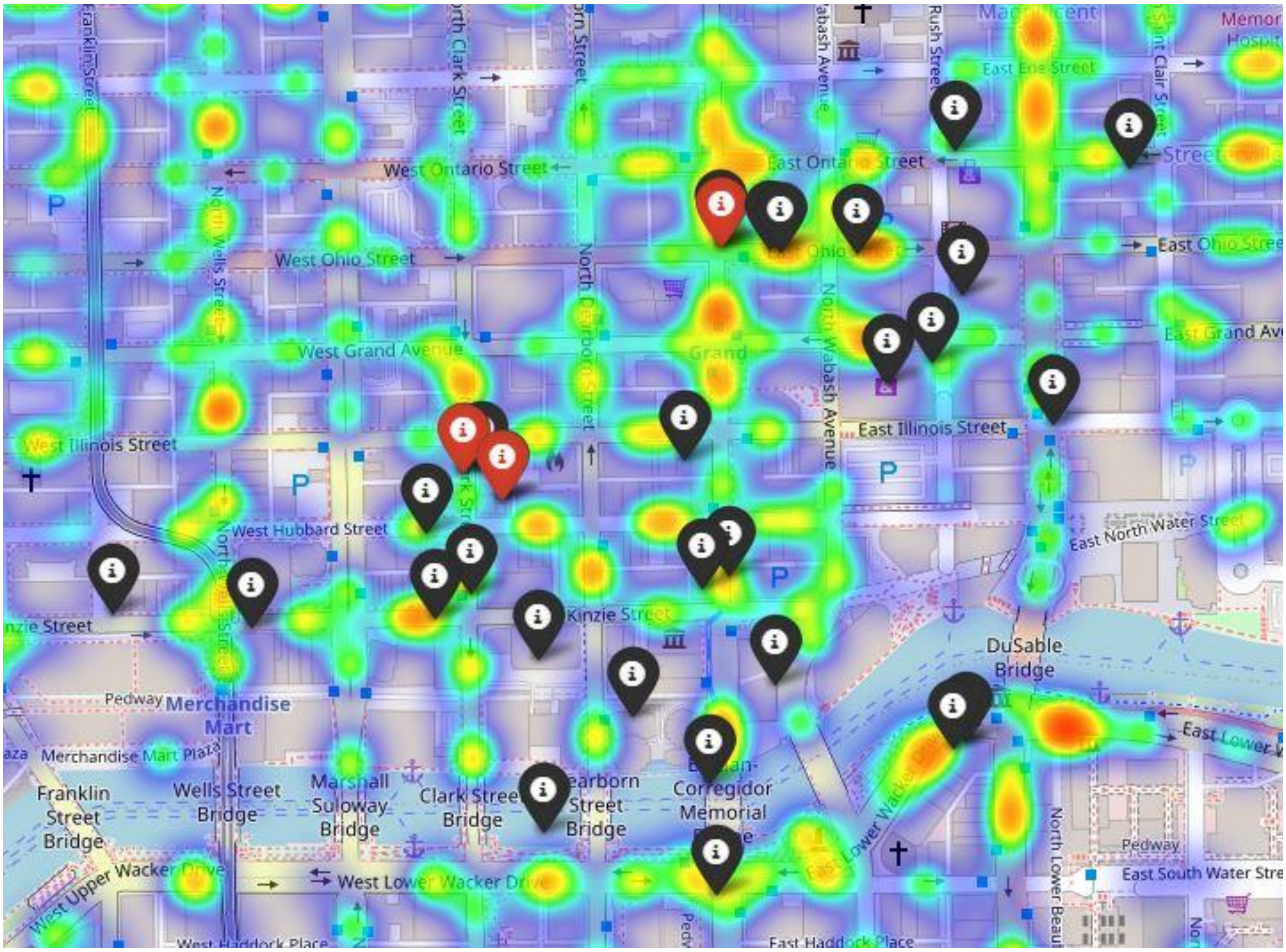
It is observed that 3 sites out of 30 have 1 in the "prediction" column.

	name	date	latitude	longitude	prediction
0	Broken Shaker	2019-12-24 22:12:00	41.892488	-87.627336	0
1	London House Rooftop Bar	2019-05-28 22:17:00	41.888031	-87.625264	0
2	Gilt Bar	2019-02-07 20:15:00	41.889236	-87.635377	0
3	The Berkshire Room	2019-08-16 05:27:00	41.892495	-87.627476	0
4	Raised Rooftop at Renaissance Hotel	2019-01-06 19:43:00	41.886726	-87.628110	0
5	Bar Sótano	2019-12-07 04:00:00	41.890518	-87.630947	0
6	La Mez Agave Lounge	2019-04-17 14:28:00	41.889196	-87.631510	0
7	The Smith	2019-01-22 16:28:00	41.889417	-87.631070	0
8	Watershed	2019-02-13 20:08:00	41.892597	-87.628059	0
9	Pops for Champagne	2019-10-07 15:08:00	41.892530	-87.628038	1
10	Three Dots and a Dash	2019-07-05 10:29:00	41.890270	-87.630690	1
11	Lobby Bar	2019-11-28 11:40:00	41.892106	-87.625147	0
12	Flora Fauna	2019-01-25 13:26:00	41.890622	-87.628495	0
13	Side Door	2019-12-24 16:45:00	41.893398	-87.625235	0
14	Barrio	2019-02-14 01:50:00	41.888826	-87.630251	0
15	Rossi's Liquors	2019-08-26 17:16:00	41.889576	-87.627965	0
16	Celeste	2019-11-10 22:59:00	41.889958	-87.631607	0
17	Enolo Wine Cafe	2019-07-17 08:28:00	41.890501	-87.631149	1
18	Copper Fox Gastropub	2019-02-14 23:58:00	41.893238	-87.623130	0
19	Foundation Room	2019-10-04 14:19:00	41.888319	-87.629124	0
20	Highline Bar + Lounge	2019-09-13 18:18:00	41.889119	-87.633716	0
21	Travelle	2019-05-08 23:35:00	41.888598	-87.627413	0
22	Birreria	2019-05-12 19:46:00	41.892466	-87.626423	0
23	Tiny Tapp	2019-08-23 14:30:00	41.887271	-87.630189	0
24	ENO	2019-08-23 23:08:00	41.890937	-87.624066	0
25	On 21	2019-12-05 14:54:00	41.888088	-87.625118	0
26	Public House	2019-05-04 04:41:00	41.889474	-87.628274	0
27	Habitant - Nordstrom Michigan Avenue	2019-06-07 19:20:00	41.891308	-87.626047	0
28	Upstairs at The Gwen	2019-10-20 19:05:00	41.891489	-87.625514	0
29	Wollensky's Grill	2019-11-22 21:08:00	41.887703	-87.628191	0

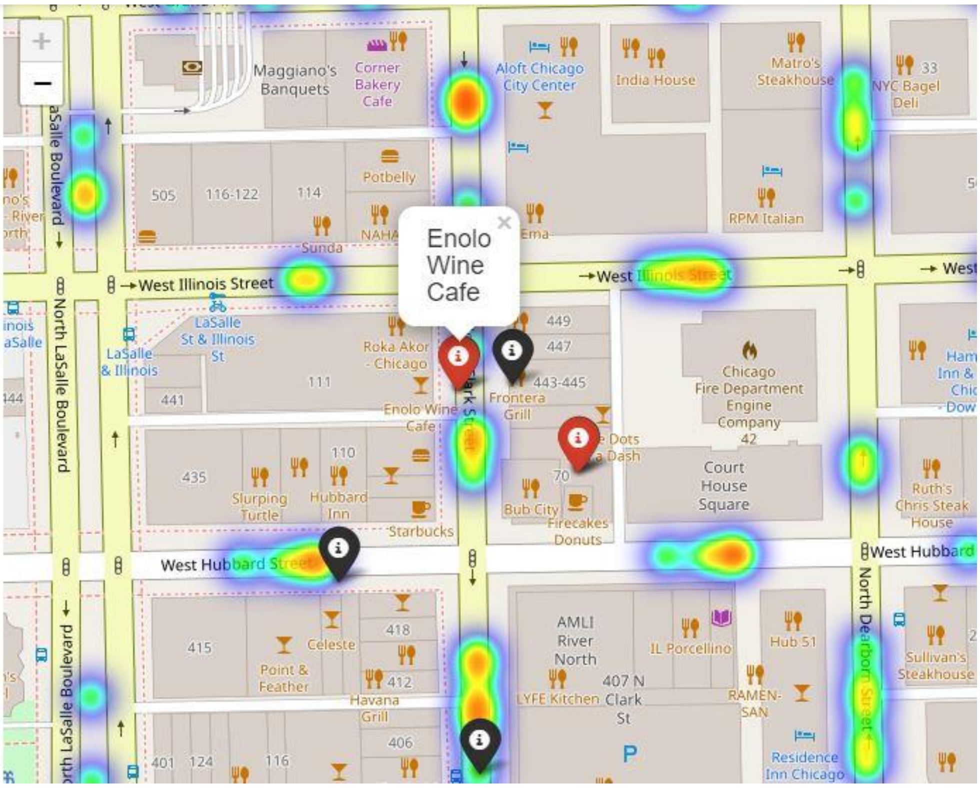
Visualisation of Predictions

1 - Enolo Wine Cafe | 2 - Three Dots and Dash | 3 - Pops for Champagne

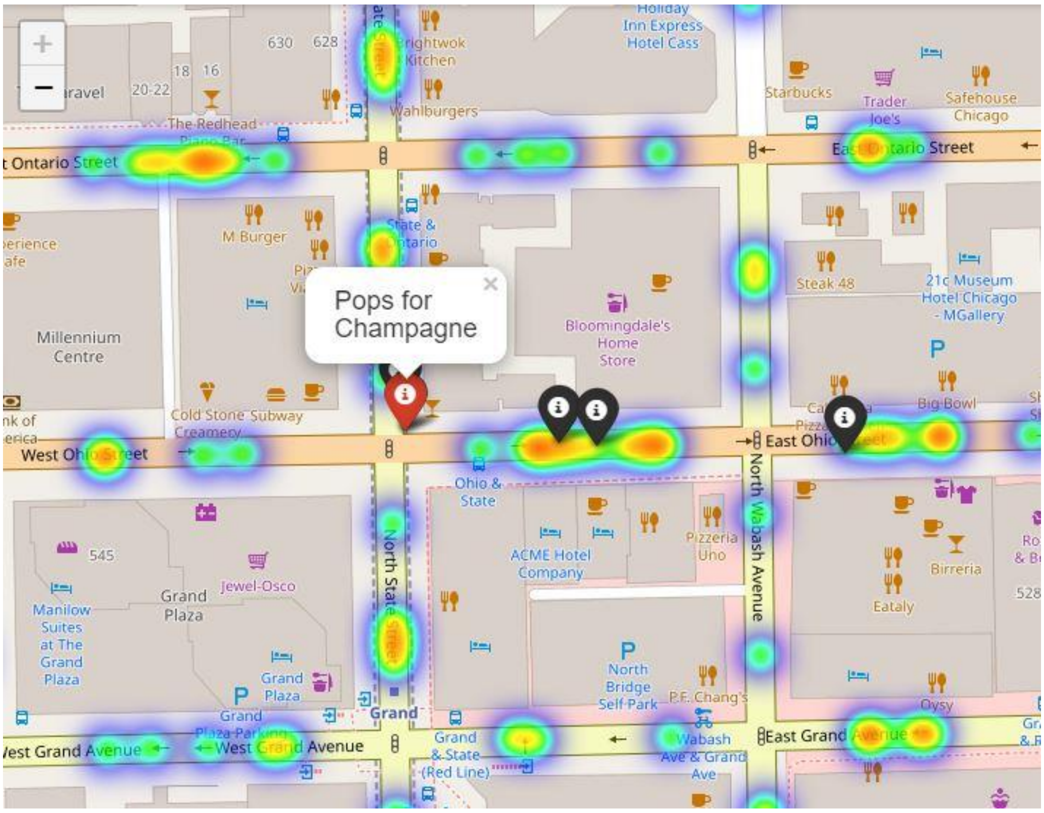
Total visualization



1 - Enolo Wine Cafe | 2 - Three Dots and Dash



3 - Pops for Champagne



Conclusion

First of all, I have to mention the growth of my skills after all the courses of the IBM Data Science Professional Certificate, all the challenges proposed were of great value for the accomplishment of this project.

The objectives were achieved despite some problems midway through. The basic plan of the Foursquare API for Developers has a very short limit of consultations per day and until I get used to it, I exceeded that limit several times, making me have to wait for the next day and later having to pass the data to my personal drive and introducing it again.

Discussion

Chicago is to be congratulated for providing open data with this quality and with an extremely interactive and simple platform, even having an area for developers.

Today (2020), we are experiencing the Coronavirus pandemic and I have observed many platforms developed for this cause, with humanitarian aid from data scientists to carry out projects on top of the data.

With the example of Chicago and the available data on Cornonavirus ... Shouldn't they look more at the importance of data and its availability?